



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**NUMERICKÉ INTEGRÁTORY V REGULÁCI**

NUMERICAL INTEGRATORS IN CONTROL THEORY

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ADAM NOVÁK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VÁCLAV ŠÁTEK, Ph.D.**

**BRNO 2017**

## Abstrakt

Táto práca sa zaoberá témou paralelného numerického riešenia parciálnych diferenciálnych rovníc so zameraním na riadenie dynamických systémov pracujúcich so vstavaným modelom riadenia v reálnom čase. Práca sa na začiatku zameriava na systémy pracujúce v reálnom čase. V ďalšej časti sa venuje analýze diferenciálnych rovníc a typom diferenciálnych rovníc. Následne rozoberá numerické derivovanie. Práca sa ďalej zaoberá jednotlivými numerickými metódami, ktoré slúžia na výpočet diferenciálnych rovníc. Práca sa tiež zaoberá jednotlivými vplyvmi, ktoré môžu ovplyvniť výpočet diferenciálnych rovníc pri Modernej metóde Taylorovho radu. Výsledkom je konkurencieschopná aplikácia umožňujúca rýchle a presné riešenie modelu riadenia v reálnom čase a následný výpočet vzniknutých diferenciálnych rovníc pomocou Modernej metódy Taylorovho radu.

## Abstract

This bachelor thesis deal with the topic of partial differential equations of parallel solutions from focus on control of dynamic systems working with real-time built-in control model. First, it is focused on real-time systems. The next section deals with the analysis of differential equations and types of differential equations. Consequently, numerical integration and derivation are discussed. The thesis deals with individual numerical methods, which are used to calculate a differential equations. This thesis also deals with individual fluids that can influence the calculation of differential equations in the Modern Taylor method. The result is a competitive application that allows a quick and accurate solution of real-time management model and the subsequent calculation of the differential equations generated using the Modern Taylor Series Method.

## Klíčové slová

diferenciálne rovnice, numerické metódy, numerické derivovanie, Taylorova rada, Moderná metóda Taylorovej rady

## Keywords

differential equations, numerical methods, numerical differentiation, Taylor series, Modern Taylor Series Method

## Citácia

NOVÁK, Adam. *Numerické integrátory v regulácii*. Brno, 2017. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Šátek Václav.

# Numerické integrátory v regulácii

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Václava Šáteka, Ph.D. Ďalšie informácie poskytol pán doc. Ing. Jiří Kunovský, CSc.. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....

Adam Novák

17. mája 2017

## Podakovanie

Moje podakovanie patrí pánovi Ing. Václavovi Šátekovi, Ph.D., za vedenie, ochotu a cenné rady poskytnuté pri vypracovaní tejto bakalárskej práce. Ďalej by som chcel poďakovať doc. Ing. Jiřímu Kunovskému, CSc. za užitočné informácie, trpezlivosť a pomoc.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Systémy v reálnom čase</b>	<b>6</b>
<b>3</b>	<b>Analýza diferenciálnych rovníc</b>	<b>8</b>
3.1	Obyčajné diferenciálne rovnice . . . . .	8
3.2	Parciálne diferenciálne rovnice . . . . .	9
<b>4</b>	<b>Numerické riešenie diferenciálnych rovníc</b>	<b>10</b>
4.1	Jednokrokové numerické metódy . . . . .	11
4.2	Taylorov rad . . . . .	11
4.2.1	Eulerova metóda . . . . .	12
4.2.2	Metóda Runge-Kutta . . . . .	13
4.3	Moderná metóda Taylorovovho radu . . . . .	14
4.4	Viac krokové numerické metódy . . . . .	15
4.4.1	Adams-Bashforthova metóda . . . . .	15
4.4.2	Adams-Moultonova metóda . . . . .	16
4.4.3	Metóda prediktor-korektor . . . . .	16
4.5	Chyby numerických metód . . . . .	16
<b>5</b>	<b>Prevod rádu</b>	<b>18</b>
5.1	Metóda znižovania rádu derivácie . . . . .	18
5.2	Metóda postupnej integrácie . . . . .	19
<b>6</b>	<b>Aplikácia BPAN</b>	<b>20</b>
6.1	Elektrický obvod . . . . .	20
6.2	Integrátorový obvod . . . . .	21
6.3	Riešenie danej sústavy . . . . .	22
6.4	Vplyv kvantovania . . . . .	25
6.5	Vplyv dopravného oneskorenia . . . . .	25
6.6	Vplyv rádu numerickej integračnej metódy . . . . .	27
6.7	Vplyv zmeny parametrov modelu riadenia . . . . .	29
6.8	Porovnanie s Matlabom . . . . .	30
<b>7</b>	<b>Záver</b>	<b>31</b>
	<b>Literatúra</b>	<b>32</b>
	<b>Prílohy</b>	<b>34</b>

<b>A</b>	<b>Kvantovanie</b>	<b>35</b>
<b>B</b>	<b>Padeho rozvoj</b>	<b>38</b>
<b>C</b>	<b>Rád metódy</b>	<b>40</b>
<b>D</b>	<b>Obsah CD</b>	<b>41</b>

# Zoznam obrázkov

2.1	Všeobecný lineárny systém[6]	6
2.2	Všeobecná stavová schéma systému[6]	7
4.1	Princíp jednokrokovej metódy [18]	12
4.2	Princíp viackrokovej metódy [18]	15
4.3	Závislosť chyby numerických metód na dĺžke kroku[18]	17
5.1	Metóda znižovania rádu derivácie	18
5.2	Metóda postupnej integrácie - bloková schéma [18]	19
6.1	Výstup elektrického obvodu	20
6.2	Vzhľad aplikácie elektrický obvod	21
6.3	Vzhľad aplikácie integrátorový obvod	23
6.4	Graf RLC obvodu	24
6.5	Padeho rozvoj pre $n=3$	27
6.6	Zvýšenie hodnoty kondenzátora na 10F	29
6.7	Zníženie hodnoty kondenzátora na 0.1F	29
A.1	Vplyv kvantovania krok = 0.05	35
A.2	Vplyv kvantovania krok = 0.1	36
A.3	Vplyv kvantovania krok = 0.5	36
A.4	Vplyv kvantovania krok = 1	37
B.1	Pahedo aproximáciá rádu $n$ v programe FOS	38
B.2	Diagonálna Pahedo aproximáciá rádu $n$ v programe FOS	39
C.1	Rozdiel výsledkov pri rádu $n$	40

# Zoznam tabuliek

4.1	Ukážka tabulkovej formy . . . . .	10
4.2	Prehľad koeficientov Adams-Bashforthovej metódy . . . . .	16
4.3	Prehľad koeficientov Adams-Moultonovej metódy . . . . .	16
6.1	Porovnanie výsledkov BPAN a FOS . . . . .	23
6.2	Porovnávanie integračných krokov u I2 resp. $i'_l$ . . . . .	25
6.3	Porovnávanie rádu metódy u I3 resp. $i'$ . . . . .	28
6.4	Porovnanie rýchlosti výpočtu v Matlabe a v aplikácii BPAN . . . . .	30

# Kapitola 1

## Úvod

V dnešnej dobe sa s elektrickými obvodmi stretávame všade. V menších rozmeroch ako mobilné telefóny, kuchynský robot a rôzne elektrické zariadenia, vo väčších rozmeroch ako súčasť mestskej hromadnej dopravy, osvetlenia a pod., alebo ich môžeme pozorovať ako vstavané systémy, ktoré využívame každodenne (elektrické spätné zrkadlo v aute). Táto práca sa zaoberá numerickými integrátormi v regulácii, ktoré určujú aké napätie prípadne aký prúd sa nachádza v danom elektrickom obvode.

Bakalársku prácu je možné rozdeliť na teoretickú a praktickú časť. Do teoretickej časti môžeme zaradiť kapitoly 2-5. V týchto kapitolách čítajúceho postupne uvedieme do témy bakalárskej práce a vysvetlíme často používané pojmy. Druhá kapitola bude venovaná systémom pracujúcim v reálnom čase. Tu bude vysvetlené aký je rozdiel v jednotlivých typoch systémov. V tretej kapitole si zanalyzujeme diferenciálne rovnice a oboznámime sa s obyčajnými a parciálnymi diferenciálnymi rovnicami. Štvrtá kapitola nám objasní numerické derivovanie a paralelne numerické metódy na riešenie diferenciálnych rovníc. Uvedieme si tu pár najvyužívanejších metód a oboznámime sa s Modernou metódou Taylorovho radu. V piatej kapitole sú opísané metódy na znižovanie rádu diferenciálnych rovníc.

V praktickej časti sa zameriame na jednotlivé vplyvy, ktoré ovplyvňujú výsledky výpočtov diferenciálnych rovníc, ich vplyvy budú ukázané na konkrétnych príkladoch. Bude tu tiež predstavená aplikácia, ktorá vznikla počas tejto práce.

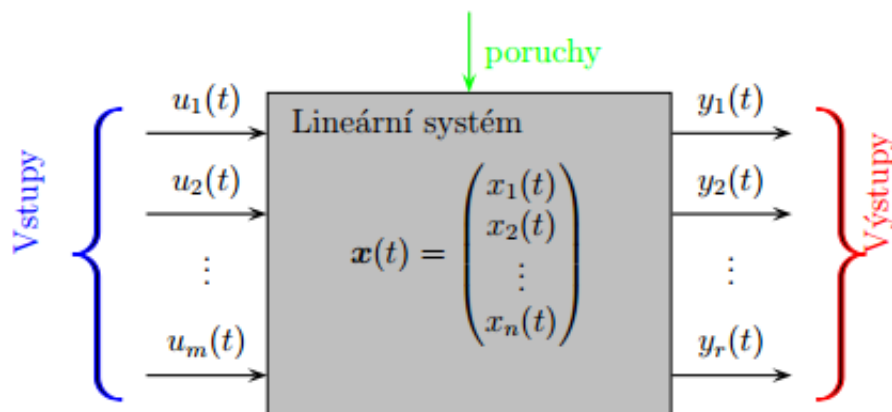


## Kapitola 2

# Systemy v reálnom čase

Táto kapitola vychádza z [14][6].

Systemom rozumieme sústavu vstupov, funkciu a sústavu výstupov. Takýto systém môžeme vidieť na obrázku 2.1. Funkcia nám opisuje aký vzťah majú vstupy a výstupy. Čas, ktorý uplynie medzi predaním vstupných hodnôt na vstup systému, vykonaním požadovaného správania systému a predaním výstupných hodnôt na výstup systému, nazývame čas odozvy systému. Rýchlosť času odozvy je závislá od charakteru a účelu špecifického systému.



Obr. 2.1: Všeobecný lineárny systém[6]

System pracujúci v reálnom čase je taký systém, ktorý spĺňa časové obmedzenia. System, ktorý nedokáže splniť jednu alebo viac požiadaviek stanovené v špecifikáciach systémových požiadaviek, je nesprávny systém. System v reálnom čase je systém, ktorého logická korektnosť je založená ako na správnosti výsledkov tak aj na ich aktuálnosti.

Systemy v reálnom čase sú reaktívne alebo vstavané.

Reaktívny systém je systém, ktorého plánovanie úloh je riadené nepretržitou interakciou (systém na kontrolu požiaru reaguje na stlačené tlačidlo).

Vstavaný systém je systém obsahujúci jeden alebo viac počítačov, ktoré majú hlavnú úlohu vo funkčnosti systému, ale systém nie je výslovne volaný počítačom.

Systemy podľa reakcie na zlyhanie poznáme:

- soft real-time systémy
- hard real-time systémy

- firm real-time systémy

Soft real-time systém je systém, v ktorom je výkon degradovaný, ale nie je zničený nedodržaním reakčne-časových obmedzení.

Hard real-time systém je systém, v ktorom pri nedodržaní čo i len jedného termínu môže viesť k úplnému zlyhaniu.

Firm real-time systém je systém, v ktorom niekoľko nedodržaných termínov nevedie k úplnému zlyhaniu, ale pri nedodržaní viacerých termínov môže viesť k úplnému zlyhaniu. Vyššie v tejto práci spomíname, že systém má sústavu vstupov a funkciu, ktorá nám tieto vstupy transformuje na sústavu výstupov. To znamená, že systém musí byť určený stavovými rovnicami. Stavové rovnice určujú teda vzťah medzi stavom systému a jeho vstupmi a výstupmi.

Prvá stavová rovnica (2.1) v maticovom tvare nám udáva vzťah medzi deriváciami stavových premenných a vektormi stavu a vstupu. Druhá stavová rovnica (2.2) v maticovom tvare určuje vzťah medzi vektorom výstupu a vektory stavu a vstupu.

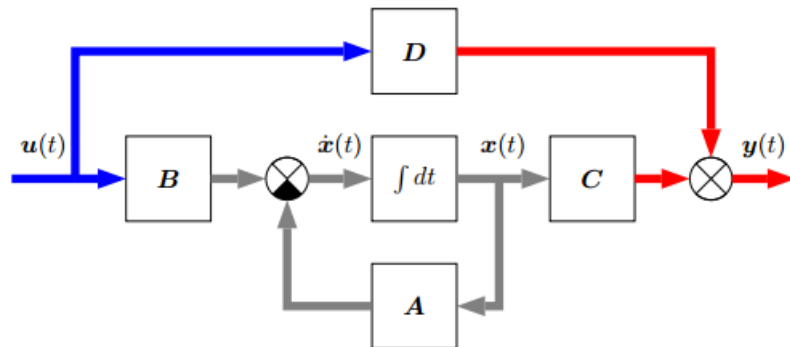
$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.1)$$

$$y(t) = Cx(t) + Du(t) \quad (2.2)$$

kde  $A$  je matica vnútorných väzieb systému,  $B$  je matica väzieb systému na vstup,  $C$  je matica väzieb výstupu na stav a  $D$  je matica priamych väzieb výstupu na vstup. Stavové vektory  $x(t)$ ,  $u(t)$  a  $y(t)$  sú stĺpcové vektory, ktoré obsahujú stavové premenné.

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} \quad u(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{pmatrix} \quad y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_r(t) \end{pmatrix}$$

Všeobecná stavová schéma je znázornená na obrázku 2.2.



Obr. 2.2: Všeobecná stavová schéma systému[6]

## Kapitola 3

# Analýza diferenciálnych rovníc

Zdroje použité v tejto kapitole [9][7][8][10][12]. Parciálnymi diferenciálnymi rovnicami sa bližšie zaoberajú publikácie [5][11][9]

Diferenciálnou rovnicou rozumieme rovnicu, v ktorej ako premenné vystupujú derivácie funkcií. Diferenciálne rovnice tvoria základy fyzikálnych výpočtov a sú používané vo väčšine oblastí ľudského poznania.

Základné rozdelenie diferenciálnych rovníc je podľa typu obsiahnutých derivácií:

- obyčajné diferenciálne rovnice (ODE) - rovnice obsahujúce derivácie len podľa jednej premennej
- parciálne diferenciálne rovnice (PDE) - obsahujú derivácie podľa viacerých premenných
- stochastické diferenciálne rovnice (SDE) - rovnice zahŕňajú najmenej jeden stochastický proces
- diferenciálne algebraické rovnice (DAE) - diferenciálne rovnice, v ktorých sa nachádzajú aj čisto algebraické vedľajšie podmienky

### 3.1 Obyčajné diferenciálne rovnice

**Definícia 3.1.1** Obyčajnú diferenciálnu rovnicu  $n$ -tého rádu nazývame rovnicu, v ktorej sa vyskytuje neznáma funkcia jednej premennej a jej derivácie až do rádu  $n$ . Zapisujeme ju všeobecne v tvare

$$F(t, y, y', \dots, y^{(n)}) = 0, \quad (3.1)$$

kde  $F$  je funkcia  $n+2$  premenných definovaná na otvorenej množine  $\Omega \subseteq \mathbb{R}^{n+2}$ . Tento tvar nazývame nerozriešený vzhľadom k najvyššej derivácii alebo implicitný tvar.

Špeciálnym prípadom rovnice (3.1) je rovnica, ktorá sa dá zapísať v tvare

$$y^{(n)} = f(t, y, y', \dots, t_{(n-1)}), \quad (3.2)$$

kde  $f$  je funkcia  $n+1$  premenných a definovaná na otvorenej množine  $\Omega \subseteq \mathbb{R}^{n+1}$ . Tento tvar rovnice nazývame rozriešený vzhľadom k najvyššej derivácii alebo explicitný tvar.

**Definícia 3.1.2** Riešením obyčajnej diferenciálnej rovnice rádu  $n$  na intervale  $I$  nazývame každou  $n$ -krát diferencovateľnú funkciu na intervale  $I$ , ktorá vyhovuje danej rovnici. Všeobecným riešením obyčajnej diferenciálnej rovnice rozumieme všeobecný predpis závislivý na

$n$  rôznych parametroch  $c_1, c_2, \dots, c_n, [c_1, c_2, \dots, c_n] \in M \subseteq \mathbb{R}^n$ , kde ľubovoľnou voľbou týchto parametrov dostaneme konkrétne riešenie, ktoré nazývame partikulárnym (čiastočným) riešením.

## 3.2 Parciálne diferenciálne rovnice

**Definícia 3.2.1** Diferenciálnu rovnicu nazývame parciálnou, pokiaľ neznáma funkcia závisí na dvoch či viacerých premenných (príslušná rovnica teda obsahuje parciálnu deriváciu tejto hľadanej funkcie). Všeobecný tvar takejto rovnice teda je

$$F(x_1, \dots, x_n, u, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_N}, \frac{\partial^2 u}{\partial x_1^2}, \dots, \frac{\partial^k u}{\partial x_N^k}) = 0; \quad (3.3)$$

kde  $u = u(x_1, \dots, x_N)$  je hľadaná funkcia. Rád najvyššej derivácie, ktorá sa v rovnici vyskytuje, potom určuje rád tejto rovnice.

**Definícia 3.2.2** Riešením rovnice (3.3) v oblasti  $\Omega \subset \mathbb{R}^N$  nazývame každú funkciu, ktorá má v  $\Omega$  spojité všetky parciálne derivácie a ktorá dosadená zároveň s týmito deriváciami do (3.3) vyhovuje pre všetky  $(x_1, \dots, x_N) \in \Omega$  tejto rovnici.

Parciálne diferenciálne rovnice je možné upravovať do iného tvaru pomocou diferenčných metód, ktorých základom je aproximácia parciálnych derivácií konečnými diferenčnými podielami. Pre kompletnosť si uvedieme, že existuje metóda sietí a metóda priamok. Metóda priamok je bližšie vysvetlená v [12].

## Kapitola 4

# Numerické riešenie diferenciálnych rovníc

Definície, vzorce a vzťahy v tejto kapitole sú prevzaté z [15][10][21][18][16][13][9]. Diferenciálne rovnice môžeme riešiť analyticky alebo numericky. V praktických problémoch sa však vyskytujú zložité rovnice, ktoré sú zadane len tabuľkovou formou 4.1, alebo ktoré je veľmi ťažké, niekedy nemožné vyjadriť analytickým spôsobom. Preto sa na výpočet využívajú numerické metódy.

$i$	0	1	...	n
$t_i$	$t_0$	$t_1$	...	$t_n$
$y_i$	$y_0$	$y_1$	...	$y_n$

Tabuľka 4.1: Ukážka tabuľkovej formy

Základnou metódou numerického derivovania je nahradiť funkciu interpolačným polynómom, prípadne inou aproximáciou a derivovať aproximujúcu funkciu. Keďže sa výpočet derivácie danej funkcie v určitom bode počíta numericky a nie analyticky, výsledok bude mať približnú hodnotu. To akú približnú hodnotu bude mať výsledok závisí na použitej numerickej metóde, s ktorou počítame.

Derivácia funkcie  $f$  v bode  $t_0$  je:

$$f'(t_0) = \lim_{h \rightarrow 0} \frac{f(t_0 + h) - f(t_0)}{h} \quad (4.1)$$

Tento vzorec poskytuje jasný spôsob, ako vytvoriť aproximáciu  $f'(x_0)$  a to:

$$f'(t_0) = \frac{f(t_0 + h) - f(t_0)}{h} \quad (4.2)$$

pre malé hodnoty  $h$ . Tento vzorec nie je vhodný, kvôli veľkej zaokrúhľovacej chybe. Na aproximáciu  $f'(t_0)$ , predpokladáme, že  $t_0 \in (a, b)$  a  $t_1 = t_0 + h$  pre  $h \neq 0$ , ktoré je dostatočne malé aby  $t_1 \in (a, b)$ . Potom môžeme zostrojiť prvý Lagrangeov polynóm  $P_{0,1}(t)$  pre  $f$  určený  $t_0$  a  $t_1$

$$f(t) = P_{0,1}(t) + \frac{f(t_1) - f(t_0)}{t_1 - t_0} f''(\xi(t)) \quad (4.3)$$

Po úpravách a odstránení termínov obsahujúcich  $\xi$  dostávame tvar

$$f'(t_0) = \frac{f(t_0 + h) - f(t_0)}{h} \quad (4.4)$$

Nevýhoda toho tvaru je, že nemáme žiadnu informáciu o tom, akej veľkej chyby sa dopúšťame pri výpočte. Preto do rovnice (4.4) dosadíme ďalšiu aproximáciu, ktorá nám objasňuje akej chyby vo výpočte sa dopúšťame.

$$f'(t_0) = \frac{f(t_0 + h) - f(t_0)}{h} - \frac{h}{2}f''(\xi) \quad (4.5)$$

V tejto práci sme si v sekcii 3.1 objasnili obyčajné diferenciálne rovnice. V práci sa ďalej budeme zaoberať diferenciálnymi rovnicami prvého rádu. Všeobecný tvar diferenciálnej rovnice prvého rádu je

$$g(t, y(t), y'(t)) = 0 \quad (4.6)$$

Predpokladajme, že rovnicu (4.6) je možné vyjadriť explicitne v tvare

$$y'(t) = f(t, y(t)) \quad (4.7)$$

Všeobecné riešenie diferenciálnej rovnice obsahuje integračnú konštantu, ktorá môže nadobúdať ľubovoľné hodnoty. Preto k jednoznačnému určeniu  $y(t)$  musíme ešte doplniť hodnotu funkcie v určitom bode  $t = t_0$ , teda počiatočnú podmienku

$$y(t_0) = y_0 \quad (4.8)$$

Potom rovnica (4.7) spolu s počiatočnou podmienkou (4.8) sa nazýva Cauchyova úloha alebo tiež počiatočná úloha.

Pri numerických metódach nehľadáme spojitú funkciu na celom skúmanom intervale  $\langle a, b \rangle$ , ale hodnoty približného riešenia počítame iba v konečnom počte bodov  $a = t_0 < t_1 < \dots < t_n = b$ . Tieto body sa nazývajú **uzlové body** alebo **uzly siete** a množine  $t_0, t_1, \dots, t_n$  hovoríme sieť. **Integračný krok** siete v uzle  $t_i$  potom bude  $h_i = t_{i+1} - t_i$ . Pokiaľ je integračný krok  $h$  konštantný medzi jednotlivými uzlami, ide o pravidelnú sieť. Metódy používajú k riešeniu rekurentný vzťah, v ktorom je  $y_{n+1}$  vyjadrená pomocou  $k$  hodnôt  $y_n, y_{n-1}, \dots, y_{n+1-k}$ . Ak je  $k = 1$ , tak hovoríme o jednokrokovej metóde. Ak je  $k > 1$ , hovoríme o viackrokovej metóde.

Diferenciálne rovnice vyššieho rádu sa musia pre numerické riešenie previesť na sústavu obyčajných diferenciálnych rovníc prvého rádu. Na tento prevod môže byť použitá:

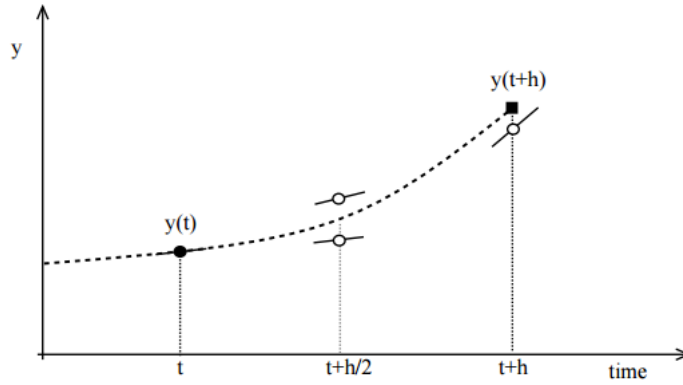
- metóda znižovania rádu derivácie, viď kapitola 5.1
- metóda postupnej integrácie, viď kapitola 5.2

## 4.1 Jednokrokové numerické metódy

Princíp jednokrokových metód je znázornený na obrázku 4.1. Derivácia v zadanom bode odpovedá smernici dotyčnice k výslednému funkčnému priebehu. Tieto smernice je možné vypočítať pre každý bod roviny podľa zadanej diferenciálnej rovnice. Keďže poznáme hodnotu riešenia  $y(t)$  na začiatku kroku, môžeme využiť hodnotu derivácie v tomto bode k výpočtu nasledujúcej hodnoty  $y(t + h)$ , kde  $h$  je integračný krok.

## 4.2 Taylorov rad

Definícia Taylorovho radu vychádza z Taylorového teorému, ktorý zostavil Brook Taylor (1685-1731). Brook Taylor radu popísal v roku 1715 v diele *Methodus incrementorum directa et inversa*.



Obr. 4.1: Princíp jednokrokovej metódy [18]

**Definícia 4.2.1** *Nech  $n \in \mathbb{N}$ ,  $t_0 \in \mathbb{R}$ ,  $f(t)$  je funkcia definovaná v okolí bodu  $t_0$ . Predpokladajme, že  $f(t)$  má v bode  $t_0$  vlastnú deriváciu do rádu  $n$  vrátane. Potom polynóm  $T_n(x)$*

$$T_n^{(t)} = \sum_{k=0}^n \frac{f^{(k)}(t_0)}{k!} (t - t_0)^k \quad (4.9)$$

*nazývame Taylorovým polynómom funkcie  $f(t)$  so stredom  $x_0$  stupňom  $n$ .*

$R_n(t)$  nazývame zvyškom po  $n$ -tom člene a určuje nám rozdiel medzi hodnotou funkcie  $f(t)$  a jej Taylorovým polynómom stupňa  $n$ .

$$f(t) = T_n^{(t)} + R_n(t) \quad (4.10)$$

**Veta 4.2.1** *Nech  $t_0, t \in \mathbb{R}$ ,  $t \neq t_0$ ,  $n \in \mathbb{N}$ . Predpokladajme, že funkcia  $f(t)$  má v uzavretom intervale  $I$  s krajnými bodmi  $t_0, t$  deriváciu až do rádu  $(n+1)$ . Nech  $T_n^{(t)}$  je určený rovnicou (4.9) a  $R_n(t)$  rovnicou (4.10). Nech  $\gamma(t)$ ,  $t \in [t_0, t]$  je funkcia spojitá na  $I$ , ktorá má v každom vnútornom bode intervalu  $I$  deriváciu  $\gamma(t) \neq 0$ . Potom existuje vnútorný bod  $\xi$  intervalu  $I$  tak, že*

$$R_n(t) = \frac{(t - \xi)^n}{n!} \frac{\gamma(t) - \gamma(t_0)}{\gamma'(\xi)} f^{(n+1)}(\xi) \quad (4.11)$$

**Definícia 4.2.2** *Nech funkcia  $f(t)$  má v bode  $t_0$  deriváciu všetkých rádov. Potom sa rada*

$$f(t) = \sum_{k=0}^{\infty} \frac{f^{(k)}(t_0)}{k!} (t - t_0)^k \quad (4.12)$$

*nazýva Taylorov rad funkcie  $f(t)$  so stredom v bode  $t_0$ .*

#### 4.2.1 Eulerova metóda

Základnou a najjednoduchšou metódou na hľadanie približného riešenia Cauchyho úlohy, je Eulerova metóda (1768-1780).

**Definícia 4.2.3** *Pokiaľ má funkcia  $y(t)$  jedinečné riešenie, to znamená, že má zadanú počiatočnú podmienku (4.8), môžeme písať*

$$y(t_{i+1}) = y(t_i) + h * f(t_i, y(t_i)) + \frac{h^2}{2} y''(\xi_i) \quad (4.13)$$

Eulerovu metódu teda môžeme zostaviť pre každé  $i = 1, 2, \dots, n$  zanedbaním chybového členu Taylorovho radu 4.2. Eulerovu metódu teda môžeme sformulovať takto:

$$y_{i+1} = y_i + h * f(t_i, y_i), \quad y_0 = y(0) \quad (4.14)$$

#### 4.2.2 Metóda Runge-Kutta

Táto metóda je tiež v podstate založená na Taylorovom rozvoji funkcie, ale nepriamo tak, aby sme nemuseli určovať vyššie derivácie funkcie, tieto sa aproximujú výpočtom samotnej funkcie vo vhodne zvolených strategických bodoch. Ich všeobecná schéma je v tvare

$$y_{n+1} = y_n + h(w_1 k_1 + \dots + w_s k_s) \quad (4.15)$$

kde

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_i &= f(t_n + a_i h, y_n + h \sum_{j=1}^{i-1} \beta_{ij} k_j), \quad i = 2, \dots, s \end{aligned} \quad (4.16)$$

kde  $w_i, a_i$  a  $\beta$  sú vhodne zvolené konštanty tak, aby mala metóda maximálny rád. Viacej o vhodne zvolených konštantách nájdete v [20].

Prvú metódu tohoto druhu navrhol Runge(1895) a ďalšiu potom Kutta (1901). Táto dvojica spoločne plne definovala metódu 4. rádu a navrhla prvú metódu 5. rádu. Metóda 4. rádu je najčastejšia používaná metóda pre riešenie numerických riešení diferenciálnych rovníc.

Runge-Kutta 2. rádu (modifikovaná Eulerova metóda)

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_1) \\ y_{n+1} &= y_n + h k_2 \end{aligned} \quad (4.17)$$

Keď sa hovorí o metóde Runge-Kutta, s najväčšou pravdepodobnosťou sa tým práve myslí Runge-Kutta 4. rádu

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_1) \\ k_3 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_2) \\ k_4 &= f(t_n, y_n + h k_3) \\ y_{n+1} &= y_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \quad (4.18)$$



### 4.3 Moderná metóda Taylorovho radu

Moderná metóda Taylorovho radu vychádza zo základnej definície Taylorovho radu 4.2. Taylorov rad si môžeme kvôli lepšej prehľadnosti prepísať do tvaru

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2!}f'(x_n, y_n) + \dots + \frac{h^n}{n!}f^{(n-1)}(x_n, y_n) \quad (4.19)$$

kde  $h$  je integračný krok,  $m$  je rád metódy a  $i$  Pri Modernej metóde Taylorovho radu je možné nastaviť presný rád metódy (ORD) podľa požadovanej presnosti výpočtu (EPS)[4][13]. Pokiaľ potrebujeme zvýšiť presnosť v jednotlivých krokoch, jednoducho zvýšime rád metódy bez nutnosti reštartovania výpočtu. Rád metódy v jednotlivých krokoch závisí na ich presnosti a tým pádom je bežné, že sa aj pri konštantných krokoch líši.

#### Princíp výpočtu

Výpočet je rozdelený na diskkrétne kroky. V každom s týchto diskrétnych krokov je vytvorených niekoľko Taylorových radov, toto závisí na počte diferenciálnych rovníc, ktoré je nutné riešiť, a tieto kroky sú vypočítané pomocou jednoduchých matematických vzťahov. Pre  $n$  diferenciálnych rovníc a rád metódy  $m$  budú mať lineárne diferenciálne rovnice tvar s počiatočnou úlohou:

$$\begin{aligned} y' &= a_{11}y + a_{12}z + \dots + a_{1n}w + b_1 & y(0) &= y_0 \\ z' &= a_{21}y + a_{22}z + \dots + a_{2n}w + b_2 & z(0) &= z_0 \\ &\vdots \\ w' &= a_{n1}y + a_{n2}z + \dots + a_{nn}w + b_n & w(0) &= w_0 \end{aligned} \quad (4.20)$$

kde  $a_{ii}, b_i$  pre  $i = 1, 2, \dots, n$  sú konštanty. Diferenciálne rovnice, je nutné na tento tvar previesť pomocou transformácií, pokiaľ nie sú v tomto tvare.

Moderná metóda Taylorovho radu počíta členy v jednotlivých krokoch až do určitej presnosti EPS alebo do maximálneho rádu metódy ORD. Hodnotu nasledujúceho kroku vypočítame pomocou súčtu všetkých vypočítaných členov v predchádzajúcom kroku a predchádzajúcu hodnotu kroku, resp. počiatočnej podmienky (v prípade  $n = 0$ )

$$\begin{aligned} y_{n+1} &= y_n + DY1_n + DY2_n + \dots + DYm_n \\ z_{n+1} &= z_n + DZ1_n + DZ2_n + \dots + DZm_n \\ &\vdots \\ w_{n+1} &= w_n + DW1_n + DW2_n + \dots + DWm_n \end{aligned} \quad (4.21)$$

kde DY, DZ a DW sú jednotlivé členy,  $m$  je rád metódy ORD potrebný pre dosiahnutie presnosti EPS. Pri výpočte prvého člena  $DY1_0$ ,  $DZ1_0$  a  $DW1_0$

$$\begin{aligned} DY1_0 &= hy' = h(a_{11}y_0 + a_{12}z_0 + \dots + a_{1n}w_0 + b_1) \\ DZ1_0 &= hz' = h(a_{21}y_0 + a_{22}z_0 + \dots + a_{2n}w_0 + b_2) \\ &\vdots \\ DW1_0 &= hw' = h(a_{n1}y_0 + a_{n2}z_0 + \dots + a_{nn}w_0 + b_n) \end{aligned} \quad (4.22)$$

kde  $h$  označuje veľkosť integračného kroku, s ktorým počítame.

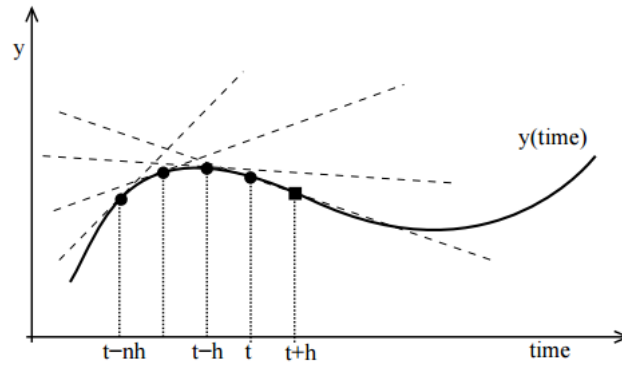
Pokiaľ požadovaná presnosť (EPS) nie je dosiahnutá a rád metódy (maxORD) taktiež nebol

dosiahnutý, tak namiesto počiatočných podmienok  $(y_0, z_0, \dots, w_0)$  použijeme novo vypočítané členy Taylorovho radu  $(DY1_0, DZ1_0, \dots, DW1_0)$  a v každom ďalšom nasledujúcom člene sa integračný krok delí aktuálnym rádom metódy  $k = 1, \dots, m$ .

$$\begin{aligned}
 DY(k+1)_0 &= \frac{h}{k+1} y^{(k+1)} = \frac{h}{k+1} (a_{11} DYk_0 + a_{12} DZk_0 + \dots + a_{1n} DWk_0) \\
 DZ(k+1)_0 &= \frac{h}{k+1} z^{(k+1)} = \frac{h}{k+1} (a_{21} DYk_0 + a_{22} DZk_0 + \dots + a_{2n} DWk_0) \\
 &\vdots \\
 DW(k+1)_0 &= \frac{h}{k+1} w^{(k+1)} = \frac{k+1}{2} (a_{n1} DYk_0 + a_{n2} DZk_0 + \dots + a_{nn} DWk_0)
 \end{aligned} \tag{4.23}$$

## 4.4 Viackrokové numerické metódy

Pr viackrokovej metóde počítame približné riešenie v ďalšom uzlovom bode siete pomocou niekoľkých predchádzajúcich uzlových bodov. Tento princíp je znázornený na obrázku 4.2. Počet použitých predchádzajúcich hodnôt závisí na rade metódy a všeobecne platí, že presnosť, ale aj výpočtová náročnosť rastie s rádom metódy.



Obr. 4.2: Princíp viackrokovej metódy [18]

Viackroková metóda má tvar

$$y_{n+1} = a_1 y_n + a_2 y_{n-1} + \dots + a_k y_{n-k} + h(b_0 f_{n+1} + b_1 f_n + \dots + b_k f_{n-k+1}) \tag{4.24}$$

kde  $k$  je prirodzené číslo a  $a_k, b_k$  sú konštanty.

Nevýhodou viackrokových metód je to, že riešenie v prvých  $k$  uzlových bodoch  $x_0, \dots, x_{k-1}$  musíme získať iným spôsobom. Na tieto uzlové body sa použijú jednokrokové metódy s rovnakým rádom presnosti, aký má ďalej použitá viackroková metóda.

Pokiaľ v rovnici (4.7) nahradíme funkciu  $f$  interpolačným polynómom s uzlami  $t_{n+1-k}, \dots, t_n$ , resp. s uzlami  $t_{n+1-k}, \dots, t_{n+1}$ , a rovnicu zintegrujeme cez interval  $\langle t_n, t_{n+1} \rangle$ , dostaneme tým explicitnú  $k$ -krokovú metódu, viď tabuľka 4.4.1, resp. implicitnú  $k$ -krokovú metódu, viď tabuľka 4.4.2.

### 4.4.1 Adams-Bashforthova metóda

Všobecný tvar Adams-Bashforthovej metódy

$$y_{n+1} = y_n + h(b_1 f_n + b_2 f_{n-1} + \dots + b_k f_{n+1-k}) \tag{4.25}$$

Prehľad koeficientov  $b_i$  pre  $k = 1, 2, 3, 4$  spolu s rádom presnosti  $p$  každej metódy

k	$b_1$	$b_2$	$b_3$	$b_4$	p
1	1				1
2	3/2	-1/2			2
3	23/12	-16/12	5/12		3
4	55/24	-59/24	37/24	-9/24	4

Tabuľka 4.2: Prehľad koeficientov Adams-Bashforthovej metódy

#### 4.4.2 Adams-Moultonova metóda

Všeobecný tvar Adams-Moultonovej metódy

$$y_{n+1} = y_n + h(b_0 f_{n+1} + b_1 f_n + \dots + b_k f_{n+1-k}) \quad (4.26)$$

Prehľad koeficientov  $b_i$  pre  $k = 0, 1, 2, 3$  spolu s rádom presnosti  $p$  každej metódy. Môžete si všimnúť, že rád  $p$  je vyšší než koeficient  $k$ .

k	$b_1$	$b_2$	$b_3$	$b_4$	p
0	1				1
1	1/2	1/2			2
2	5/12	8/12	-1/12		3
3	9/24	19/24	-5/24	1/24	4

Tabuľka 4.3: Prehľad koeficientov Adams-Moultonovej metódy

#### 4.4.3 Metóda prediktor-korektor

Princíp metódy: V každom kroku vypočítame počiatočnú aproximáciu  $y_{n+1}^{(0)}$  pomocou explicitnej viackrokovej metódy - prediktoru. Túto hodnotu spresníme použitím implicitnej viackrokovej metódy - korektoru. Niekedy sa korektor použije raz, ale je možné, že sa bude iterácia opakovať, pokiaľ nedosiahne určitú presnosť. Pri dvojici prediktor-korektor volíme vždy explicitnú a implicitnú metódu rovnakého rádu.

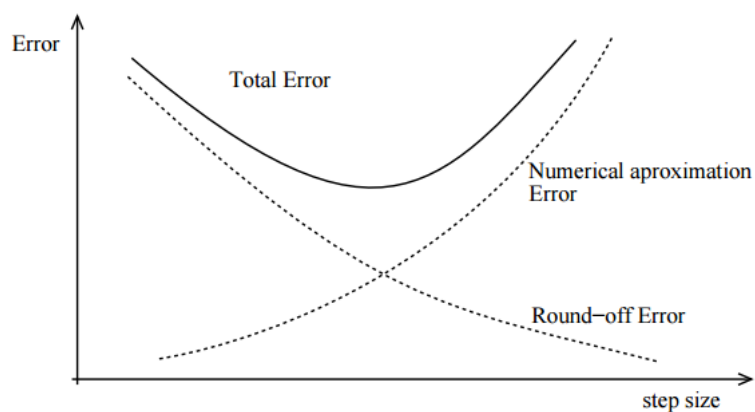
### 4.5 Chyby numerických metód

Najdôležitejšou vlastnosťou numerických metód je ich presnosť. Časová náročnosť je ďalšou nevyhnutnou vlastnosťou numerických metód, na ktorú je nutné brať zreteľ. Pokiaľ počítame jednoduché diferenciálne rovnice, čas, ktorý potrebuje numerická metóda na výpočet, ani nepostrehneme. Avšak ak počítame komplexné úlohy s viacerými diferenciálnymi rovnicami, tak časová náročnosť výpočtu hrá veľkú rolu. Ďalšou dôležitou vlastnosťou je stabilita numerického riešenia pri použití danej metódy.

Výsledná chyba je daná súčtom niekoľkých vplyvov:

- Lokálna chyba
  - zaokrúhľovacia chyba je závislá na presnosti aritmetiky počítača

- chyba metódy numerickej aproximácie je daná predovšetkým použitým rádom metódy
- Akumulovaná chyba - spôsobená sčítaním vplyvov lokálnych chýb, rastie v priebehu výpočtu



Obr. 4.3: Závislosť chyby numerických metód na dĺžke kroku[18]

## Kapitola 5

# Prevod rádu

V tejto kapitole sa budeme zaoberať pojmami, ktoré boli použité v predchádzajúcich kapitolách. Na vzťahy, obrázky a definície v tejto kapitole boli využité tieto zdroje [18][22].

### 5.1 Metóda znižovania rádu derivácie

Metódu znižovania rádu derivácie môžeme použiť pre isté špeciálne ako lineárne tak nelineárne diferenciálne rovnice vyššieho rádu. Pokiaľ daná diferenciálna rovnica neobsahuje člen  $y$  ale len členy  $y', y'', \dots$ , tak môžeme zaviesť substitúciu  $y'(x) = z(x)$  do danej diferenciálnej rovnice dostaneme diferenciálnu rovnicu pre neznámu funkciu  $z$  rádu o 1 menšieho ako bol rád pôvodnej rovnice. Pokiaľ túto rovnicu vieme riešiť a získame jej riešenie  $z(x)$ , tak integráciou získame riešenie  $y(x)$  pôvodnej rovnice. Podobne, pokiaľ daná diferenciálna rovnica neobsahuje členy  $y, y'$  ale iba členy  $y'', y''', \dots$ , potom zavedením substitúcie  $y''(x) = z(x)$  do danej diferenciálnej rovnice dostaneme diferenciálnu rovnicu pre neznámu funkciu  $z$  rádu o 2 menšieho než bol rád pôvodnej rovnice.

Metóda znižovania rádu derivácie nesmie obsahovať deriváciu vstupov.

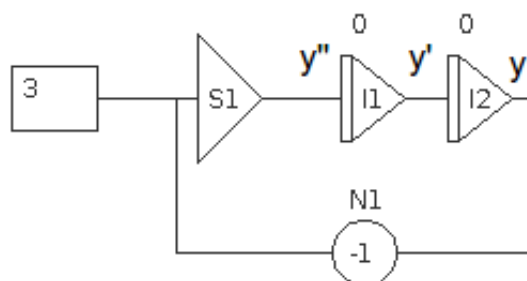
Príklad:

$$y'' + y - 3 = 0$$

$$y'' = 3 - y$$

$$y' = \int 3 - y$$

$$y = \int \int 3 - y$$



Obr. 5.1: Metóda znižovania rádu derivácie

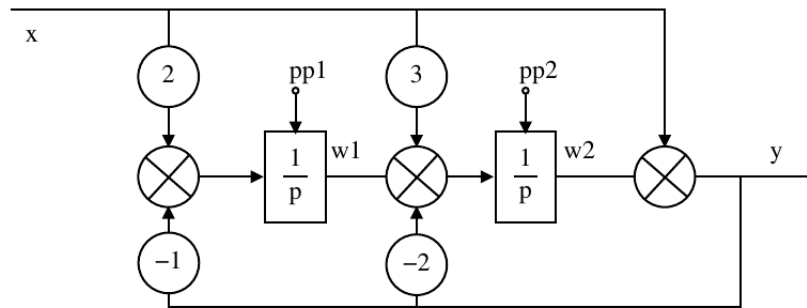
## 5.2 Metóda postupnej integrácie

Metóda postupnej integrácie je vhodná pre rovnice s deriváciami vstupov na pravej strane. Pri zadanej rovnici musíme osamostatniť najvyšší rád derivácie. Rovnicu ďalej postupne integrujeme a zavádzame nové stavové premenné. Pri každej integrácii zavádzame nové počiatočné podmienky.

Príklad:

$$\begin{aligned}y'' + 2y' + y &= x'' + 3x' + 2x \\p^2 &= p^2x + p(3x - 2) + (2x - y) \\py &= px + (3x - 2y) + \frac{1}{p}(2x - y), \quad w_1 = \frac{1}{p}(2x - y) \\y &= x + \frac{1}{p}(3x - 2y + w_1), \quad w_2 = \frac{1}{p}(3x - 2y + w_1)\end{aligned}$$

Schéma zapojenia tejto rovnice bude vyzeráť nasledovne



Obr. 5.2: Metóda postupnej integrácie - bloková schéma [18]

a výsledná sústava teda bude mať tvar

$$\begin{aligned}w_1 &= \frac{1}{p}(2x - y), \quad pp1 = 0 \\w_2 &= \frac{1}{p}(3x - 2y + w_1), \quad pp2 = 0 \\y &= x + w_2\end{aligned}$$

## Kapitola 6

# Aplikácia BPAN

Aplikácia BPAN je software vytvorený počas tvorby tejto bakalárskej práce. V jednej časti slúži ako zjednodušenie elektrického obvodu a následný výpis diferenciálnych rovníc obvodu. V druhej časti prevádza model riadenia v reálnom čase na diferenciálne rovnice a následne ich rieši pomocou Modernej metódy Taylorovho radu.

Na implementovanie zdrojových kódov aplikácie BPAN som si zvolil programovací editor Netbeans [3] v jazyku Java. Jazyk Java je objektovo orientovaný programovací jazyk. Veľkou výhodou jazyka Java je jeho prenositeľnosť medzi platformami.

Ovládanie celej aplikácie je intuitívne a jednoduché. Po drobných úpravách je možné aplikáciu využiť na výukové účely v predmetoch IEL a VNV. Program sa delí na dve časti:

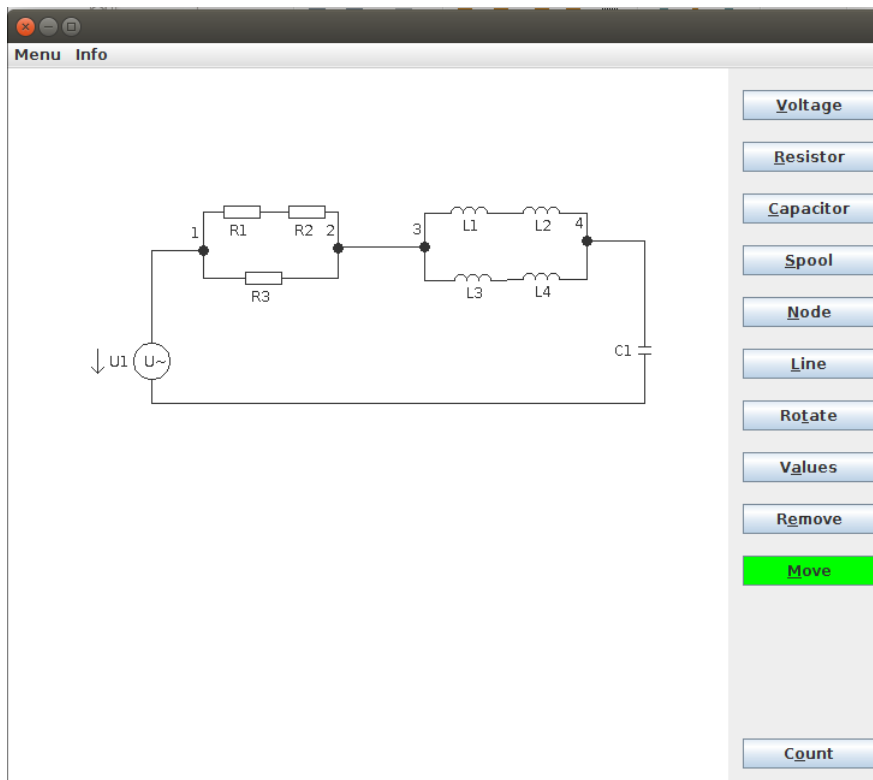
- Elektrický obvod
- Integrátorový obvod

### 6.1 Elektrický obvod

Táto časť aplikácie BPAN pracuje s elektrickými obvodmi. Z jednotlivých elektrických obvodov odstráni sériovo a paralelne zapojené komponenty (napr. dva sériovo zapojené rezistory). Následne vypíše diferenciálne rovnice daného obvodu. Aplikácia zatiaľ nemá implementované metódy, ktoré by riešili zapojenie komponentov do trojuholníka a do hviezdy. Preto je nutné kresliť elektrický obvod bez týchto druhov zapojení. Po nakreslení elektrického obvodu a prípadnom dosadení jednotlivých hodnôt do komponentov, aplikácia po stlačení tlačidla „Count“ vypíše diferenciálne rovnice daného obvodu. Výzor aplikácie s elektrickým obvodom je znázornený na schéme 6.2. Výstup elektrického obvodu zo schémy 6.2 je znázornený na schéme 6.1.

```
R2 = R2 + R1
L2 = L2 + L1
L4 = L4 + L3
R2 = R2 * R3 / (R2 + R3)
L2 = L2 * L4 / (L2 + L4)
Diferenciálne rovnice:
uc' = (1/C)*il
il' = (1/L)*(u-uc-(R*il))
```

Obr. 6.1: Výstup elektrického obvodu



Obr. 6.2: Vzhľad aplikácie elektrický obvod

V aplikácii je implementovaná funkcia, ktorá nám pomáha nájsť pravý strom v elektrickom obvode a tým pádom nám zistí, či je možné daný elektrický obvod riešiť. Je v nej tiež implementovaný algoritmus BFS (prehľadávanie do šírky). Algoritmus BFS je mierne modifikovaný a pomáha pri vyhľadávaní najkratších ciest v elektrickom obvode, čiže vypisuje najkratšie možné formy diferenciálnych rovníc. Taktiež tu je implementovaná metóda uzlových napätí a slučkových prúdov, ktorá pomáha aj s vypisovaním diferenciálnych rovníc.

## 6.2 Integrátorový obvod

Táto časť aplikácie BPAN pracuje so schémou modelu riadenia v reálnom čase. Z jednotlivých schém vytvorí diferenciálne rovnice, ktoré následne vypočíta pomocou Modernej metódy Taylorovho radu[13]. Túto metódu sme si objasnili v sekcii 4.3. Keďže je nutné, aby výpočet bol čo najpresnejší, používame v aplikácii viacslovnú aritmetiku. V jazyku Java sme na to využili triedu BigDecimal. BigDecimal je nezmeniteľné, ľubovoľne presné desatinné číslo. BigDecimal pozostáva z ľubovoľne presnej celočíslenej hodnoty a 32-bitovej celočíslenej stupnice. Viac o BigDecimal [1]. Premenné BigDecimal majú prednastavenú hodnotu presnosti na 10.

Pri kreslení schémy možno využiť tieto komponenty:

- Sčítacia (Sx) -  $i$  vstupov a  $n$  výstupov

$$S_{out} = \sum_{k=1}^i S_{in} k$$



- Násobička (Nx) -  $N_{out} = N_{in} * \text{hodnota } N$ , 1 vstup a 1 výstup
- Integrátor (Ix) - integruje hodnotu  $I = \int I'$ , 1 vstup a  $n$  výstupov

kde  $x$  jednoznačne určuje komponenty,  $i$  počet vstupov do komponentov a  $n$  je počet výstupov z komponentov. Na nakreslenie schémy môžeme využiť výstup z prvej časti aplikácie, viď obrázok 6.1. Prevod diferenciálnych rovníc v obrázku 6.1 na schému:

- $N2 \Rightarrow N2_{out} = (u - uc - (R * il)) * 1/L$ , kde  $L$  sa rovná  $1H$ ,
- $N3 \Rightarrow N3_{out} = il * 1/C$ , kde  $C$  sa rovná  $1F$ ,
- $N4 \Rightarrow N4_{out} = uc * (-1)$ ,
- $N5 \Rightarrow N5_{out} = il * R$ , kde  $R$  sa rovná  $1 \Omega$ ,
- $I3 \Rightarrow I3_{out} = il, \quad I3_{in} = il' = (1/L) * (u - uc - (R * il)), \quad il'(0) = 0$ ,
- $I4 \Rightarrow I4_{out} = uc, \quad I4_{in} = uc' = (1/C) * il, \quad uc'(0) = 0$ ,
- $S1 \Rightarrow S1_{out} = u - uc - (R * il)$ ,
- $I1, I2$  a  $N1$  si popíšeme nižšie.

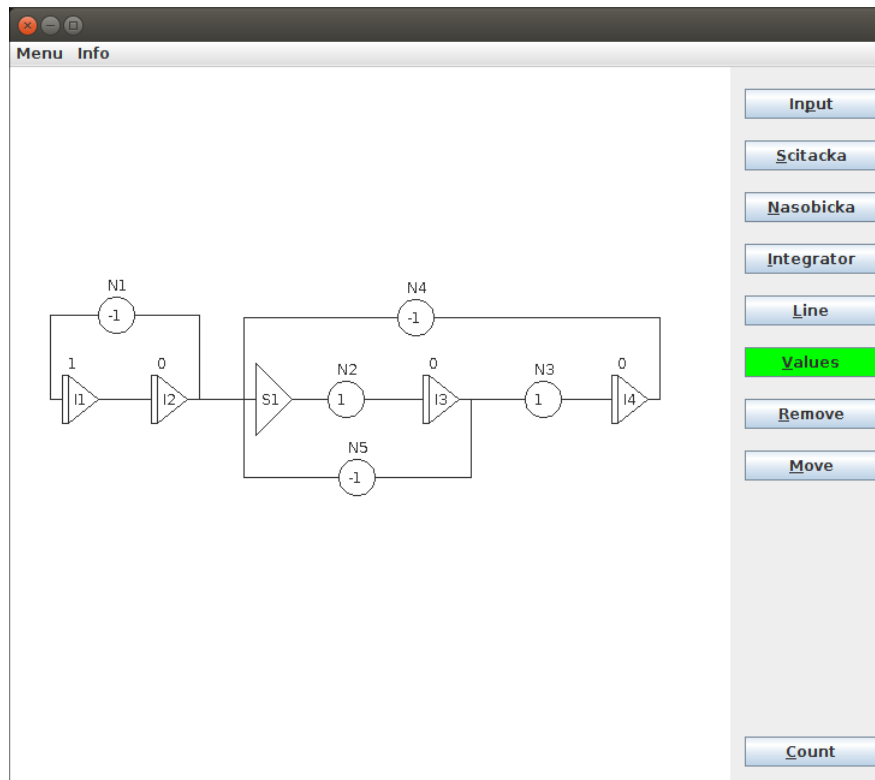
V aplikácii je vstup modelovaný ako konštanta, avšak v elektrických obvodoch býva častejšie striedavý prúd. Striedavý prúd nám opisuje sínusoida, ktorú môžeme zobrazit dvoma zapojenými integrátormi. Je nutné aby boli počiatočné podmienky integrátorov  $I1(0) = 1$  a  $I2(0) = 0$ . Tým pádom nám  $I2$  vykresľuje sínusoidu, zatiaľ čo  $I1$  nám vykresľuje kosínusoidu.

Po nakreslení schémy modelu riadenia v reálnom čase, aplikácia stlačením tlačidla „Count“ vypíše diferenciálne rovnice danej schémy (môžeme si overiť či sme zostrojili obvod správne, prípadne, ak sme vytvorili schému z vlastnej hlavy, diferenciálne rovnice schémy nám budú užitočné). Následne tieto diferenciálne rovnice vypočíta a zobrazí graf.

### 6.3 Riešenie danej sústavy

Po nakreslení schémy a zvolení tlačidla „Count“ nám vyskočí dialógové okno, ktoré sa nás pýta na veľkosť kroku ( $dt$ ), maximálny čas výpočtu ( $tmax$ ), presnosť výpočtu ( $eps$ ), maximálny rád ( $maxORD$ ) a voľiteľný parameter „precision“, ktorý určuje rozsah premenných a tým pádom aj výpočtu. Pokiaľ potrebujeme zväčšiť presnosť výpočtu je nutné zvýšiť nie len parameter  $eps$ , ale aj parameter „precision“, touto zmenou sa nám však výpočet spomaľuje. Okno obsahuje ďalšie dva parametre, ktoré si rozoberieme v sekcii 6.5.

Ak si zoberieme diferenciálne rovnice z obrázku 6.1 a na vstup privedieme jednosmerný prúd, teda  $u = 1$ , a po zadaní jednotlivých parametrov ( $dt = 0.1$ ,  $tmax = 15$ ,  $eps = 1e-30$ ,  $maxORD = 30$ ) nám aplikácia zobrazí graf 6.4.



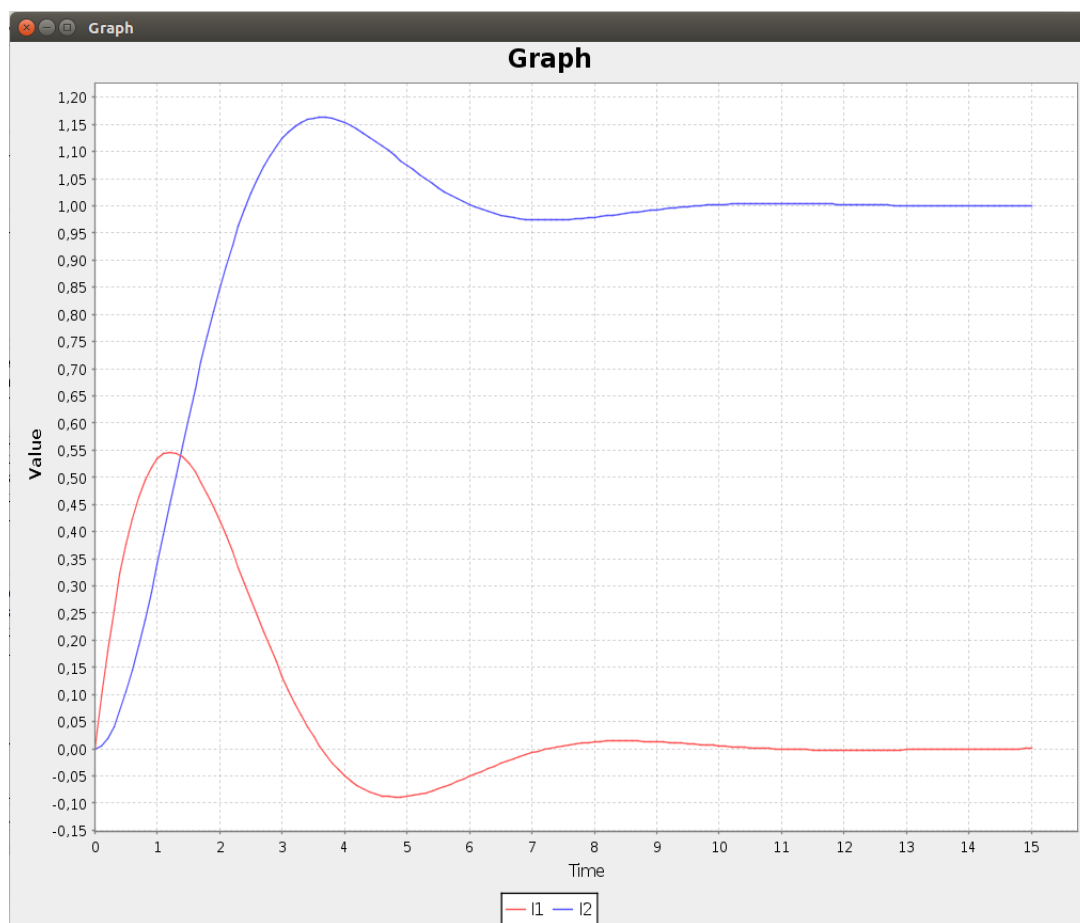
Obr. 6.3: Vzhľad aplikácie integrátorový obvod

Aby sme sa presvedčili, že implementovaná metóda v aplikácii BPAN počíta správne môžeme si niekoľko výsledkov porovnať s programom FOS [2] 6.1. V tabuľke vidíme, že

t	BPAN		FOS	
	$u'_c$	$i'$	$u'_c$	$i'$
0.1	0.0048334152	0.0950040834	0.0048334152	0.0950040833
5.0	1.0745905667	-0.0879424207	1.074590567	-0.0879424207
10.0	1.0021701167	0.0053854806	1.002170117	0.0053854806
15.0	0.9993645176	0.0002627515	0.9993645176	0.0002627514

Tabuľka 6.1: Porovnanie výsledkov BPAN a FOS

implementovaná Moderná metóda Taylorovho radu počíta v aplikácii BPAN presne. Pokiaľ by sme chceli, aby boli výsledky ešte presnejšie stačí zvýšiť premennú EPS a premennú „precision“, čo bude mať za následok spomalenie výpočtu.



Obr. 6.4: Graf RLC obvodu

## 6.4 Vplyv kvantovania

Vzhľadom k tomu, že počítače a ďalšie zariadenia spracovávajú digitálny signál a vedia vyjadriť čísla len s obmedzenou presnosťou, je potrebné navzorkované hodnoty upraviť aj na zvislej osi. Keďže v našej práci je kvantovanie vyjadrené hodnotou integračného kroku, ukážeme si príklad so zmenou veľkosti integračného kroku. V nasledujúcej tabuľke sme využili diferenciálne rovnice z obrázku 6.1 a využili sme jednosmerný prúd, čiže  $u = 1$ . Uvádzame tu použité vstupné parametre, kvôli upresneniu výpočtu  $\text{eps} = 1\text{e-}10$ ,  $\text{maxORD} = 50$ ,  $\text{precision} = 10$ ,  $\text{tmax} = 15$  a  $h = 0.05, 0.1, 0.5, 1$ . Keďže sme mali parametre  $\text{eps}$

h	0.05	0.1	0.5	1
čas[ms]	1589	311	27	14
ORD	6,7,8	7,8,9	10,11,12	13,14,15

Tabuľka 6.2: Porovnávanie integračných krokov u I2 resp.  $i_l'$

a „precision“ rovnaké pri všetkých integračných krokoch, výsledky boli totožné (do úvahy sme brali 10 cifier). Avšak maximálny rád nám stúpал spolu so zväčšovaním integračného kroku. Naopak pri znižovaní integračného kroku nám stúpал čas, strávený výpočtom. Grafy vytvorené z jednotlivých integračných krokov nájdete v prílohe A. V grafoch môžeme vidieť, že pri znižovaní integračného kroku sa nám vykreslenie vyhladzuje. Pri veľkosti integračného kroku 0.05 a 0.1 je tento rozdiel nepatrný, preto je v našom prípade vhodnejšie uskutočniť výpočet pomocou integračného kroku 0.1 vzhľadom na čas, ktorý algoritmus potrebuje na výpočet.

## 6.5 Vplyv dopravného oneskorenia

Vzťahy a rovnice v tejto sekcii sú prevzaté z [19][17]. V tejto práci sa zaoberáme dopravným oneskorením Padeho rozvojom.

Padeho aproximácia je metóda reprezentujúca jednu z najobľúbenejších a najpoužívanějších aproximácií. Je založená na porovnávaní derivátov aproximujúcich a aproximovaných funkcií v nule. Padeho aproximácia je daná nasledujúcim vzťahom

$$\exp(-pT) \approx \frac{P(-p)}{P(p)}$$

$$P(p) = \sum_{k=0}^n \binom{n}{k} \frac{2n-k}{!} 2n(pT)^k \quad (6.1)$$

kde  $n$  je rád metódy,  $p$  je Laplaceov operátor a  $T$  je požadované oneskorenie.

A metóda nazvaná diagonálna Padeho aproximácia môže byť vyjadrená vzťahom

$$\exp(-pT) \approx \frac{P(p)}{P(-p)}$$

$$P(p) = \sum_{k=0}^n \frac{2n-k}{!} k!(n-k)!(-pT)^k \quad (6.2)$$

kde  $n$  je rád metódy,  $p$  je Laplaceov operátor a  $T$  je požadované oneskorenie.

Vidíme, že nám Padeho oneskorenie aproximuje až do členu  $n$ . Pri obmedzení nekonečného

radu sa dopúšťame určitej nepresnosti. Avšak aproximácia Padeho polynómov nám tieto chyby minimalizuje vhodnou voľbou koeficientov pri jednotlivých Laplaceových operátoroch. Tieto hodnoty koeficientov závisia na rade aproximačného polynómu. Napr. pre  $n = 3$  má rovnica (6.2) tvar

$$\exp(-pT) = \frac{y}{z} = \frac{120 - 60pT + 12(pT)^2 - (pT)^3}{120 + 60pT + 12(pT)^2 + (pT)^3} \quad (6.3)$$

Rovnicu (6.4) musíme upraviť na sústavu diferenciálnych rovníc prvého rádu.

$$120z - 24pTz + 12(pT)^2z - (pT)^3z = 120y - 24pTy + 12(pT)^2y - (pT)^3y \quad (6.4)$$

Pri vstupnej funkcii  $z$  sa nám nachádzajú derivácie a tým pádom nemôžeme pre úpravu použiť metódu znižovania rádu derivácie 5.1. Musíme preto použiť inú metódu a to:

- metóda znižovania rádu derivácie s pomocnou premennou
- metóda postupnej integrácie

Pri metóde postupnej integrácie je nutné prepočítavanie počiatočných podmienok, preto použijeme metódu znižovania rádu derivácie s pomocnou premennou.

Z rovnice (6.4) môžeme vyjadriť výstup systému

$$y = z \frac{120 - 60pT + 12(pT)^2 - (pT)^3}{120 + 60pT + 12(pT)^2 + (pT)^3} \quad (6.5)$$

Zavedieme pomocnú premennú  $v$  z rovnice (6.5)

$$v = \frac{z}{120 + 60pT + 12(pT)^2 + (pT)^3} \quad (6.6)$$

Rovnicu (6.5) prepíšeme na základný tvar

$$y = 120v - 60pTv + 12(pT)^2v - (pT)^3v \quad (6.7)$$

Rovnicu (6.6) prepíšeme do tvaru vhodného pre metódu znižovania rádu derivácie a získanú diferenciálnu rovnicu upravíme

$$\begin{aligned} z &= 120v + 60pTv + 12(pT)^2v + (pT)^3v \\ (pT)^3v &= z - 120v - 60pTv - 12(pT)^2v \quad /T^3 \\ p^3v &= \frac{z}{T^3} - \frac{120v}{T^3} - \frac{60pv}{T^2} - \frac{12p^2v}{T} \end{aligned} \quad (6.8)$$

Použijeme metódu znižovania rádu derivácie

$$\begin{aligned} p^2v &= \frac{1}{p}p^3v - pp(0) \\ p^1v &= \frac{1}{p}p^2v - pp(0) \\ v &= \frac{1}{p}p^1v - pp(0) \end{aligned} \quad (6.9)$$

Všetky rovnice majú počiatočnú podmienku rovnú 0 a zlomok  $\frac{1}{p}$  reprezentuje integrátor. Z rovníc (6.8) a (6.5) môžeme koeficienty nahradiť premennými

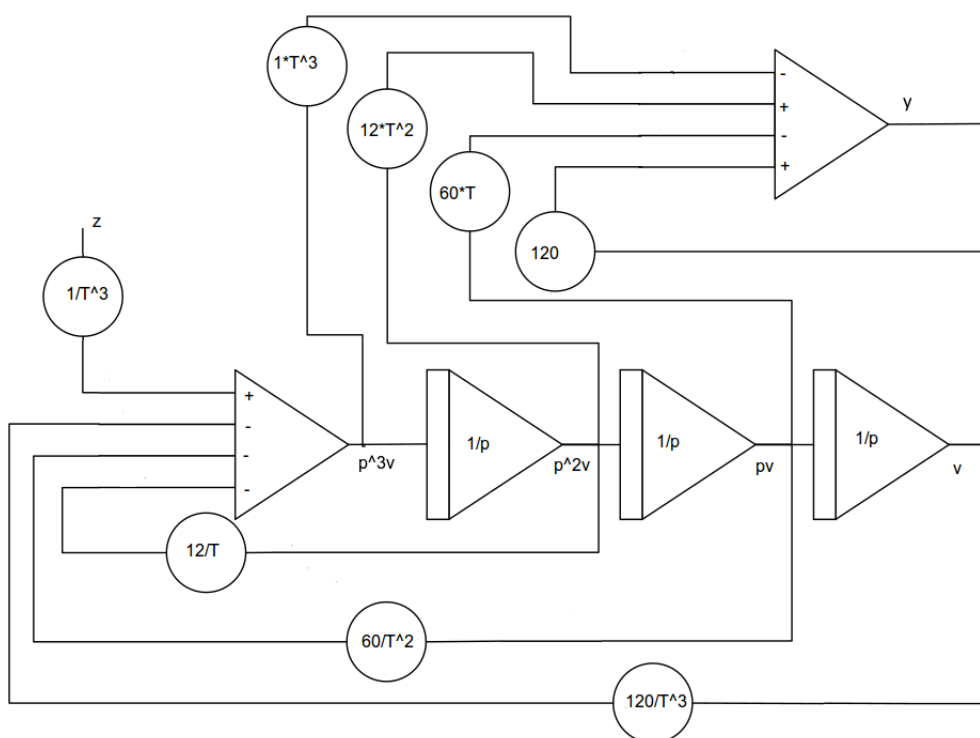
$$k_0 = 120$$

$$k_1 = 60$$

$$k_2 = 12$$

$$k_3 = 1$$

V blokovej algebre môžeme sústavu rovníc a po prevedení substitúcie reprezentovať pomocou schémy 6.5



Obr. 6.5: Padeho rozvoj pre  $n=3$

Aplikácia BPAN nám po zadaní parametrov Pade čas a Pade rád vypíše sústavu diferenciálnych a lineárnych rovníc, ktorú vieme využiť v programe TKSL alebo FOS. Po pokuksoch s Padeho rozvojom sme zistili, že obyčajný Padeho rozvoj sa pri rádu 5 a vyšším stáva nestabilným. Na základe týchto zistení sme v aplikácii implementovali algoritmus na výpočet koeficientov pomocou diagonálnej Padeho aproximácie, ktorá so zvyšovaním rádu vyhladzuje čas oneskorenia. Tieto pokusy nájdete v prílohe B.

## 6.6 Vplyv rádu numerickej integračnej metódy

Pokiaľ v danom integračnom kroku výpočet nedosiahne požadovanú presnosť (EPS) medzi dvomi za sebou nasledujúcimi členmi, výpočet ukončí zadaný rád metódy. Preto je dôležité zvoliť vhodný rád metódy, aby neukončil výpočet predčasne, naopak aby pri veľkej alebo nedosiahnuteľnej presnosti výpočet ukončil. V algoritme máme nemennú veľkosť integračného kroku (tzn. akú hodnotu integračného kroku  $h$  resp. v aplikácii  $dt$  si zvolíme, takú hodnotu

bude používať počas celého výpočet). Ako príklad si zvolíme schému 6.3 s tým, že vstup bude  $u = 1$ . Keďže vstup integrátoru I3 resp.  $i'$  závisí na hodnotách z výstupu integrátoru I3 resp.  $i$  a I4 resp.  $u_c$ , tým pádom sa nám chyba výpočtu lepšie prejaví, zvolíme si ho za pozorovaný integrátor. V tabuľke 6.4 môžete vidieť hodnoty v čase so zmenou rádu. Kvôli prehľadnosti sme tu vypísali len časť výsledok pre porovnaní. Pre upresnenie tu uvádzame použité hodnoty vo výpočtoch  $dt = 0.1$ ,  $t_{\max} = 15$ ,  $\epsilon = 1e-10$  a  $\max\text{ORD} = 10, 5$  a  $3$ .

t/maxORD	10	5	3
1.0	0.5335071951	0.5334996640	0.5684915501
2.0	0.4192796297	0.4193008986	0.4359691034
3.0	0.1332426440	0.1332760294	0.1129595136
4.0	-0.0495298797	-0.0495105446	-0.0831461490
5.0	-0.0879424207	-0.0879438960	-0.1077520034
6.0	-0.0508923182	-0.0509043792	-0.0502552215
7.0	-0.0076437137	-0.0076540127	0.0034202622
8.0	0.0127150956	0.0127117613	0.0221931498
9.0	0.0128046712	0.0128065945	0.0156877455
10.0	0.0053854806	0.0053886934	0.0033883668
11.0	-0.0004781632	-0.0004762504	-0.0035105646
12.0	-0.0023569927	-0.0023567712	-0.0040116930
13.0	-0.0016764380	-0.0016770809	-0.0017094471
14.0	-0.0004504119	-0.0004510512	0.0002512642
15.0	0.0002627515	0.0002624861	0.0008583776

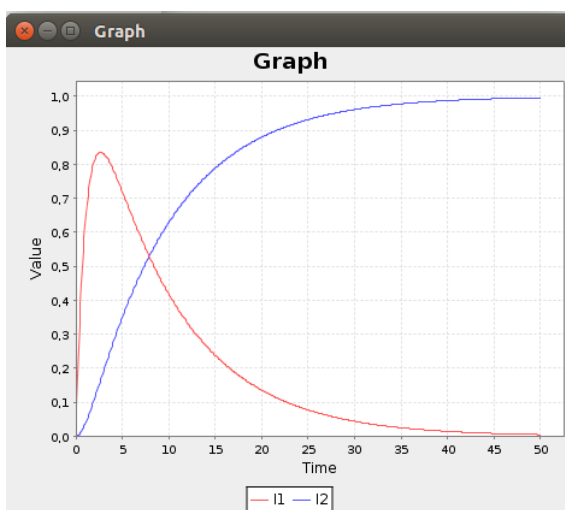
Tabuľka 6.3: Porovnávanie rádu metódy u I3 resp.  $i'$

Podľa tabuľky 6.4 a grafu v prílohe C si môžeme všimnúť do akej miery nám nevhodne zvolený parameter  $\max\text{ORD}$  ovplyvní výsledky. Pokiaľ by sme chceli tieto nepresnosti zmenšiť inak ako  $\max\text{ORD}$ , môžeme využiť zmenšenie integračného kroku, čo by malo za následok spomalenie výpočtu viď sekcia 6.4. Pre zaujímavosť, ak by sme chceli aby mal výpočet s parametrom  $\max\text{ORD} = 5$  rovnaké výsledky (s ohľadom na 10 zobrazených cifier) ako výpočet s parametrom  $\max\text{ORD} = 10$  je nutné zmenšiť integračný krok z  $h = 0.1$  na  $h = 0.01$ . Týmto bude výpočet namiesto 0,337s trvať 11s.

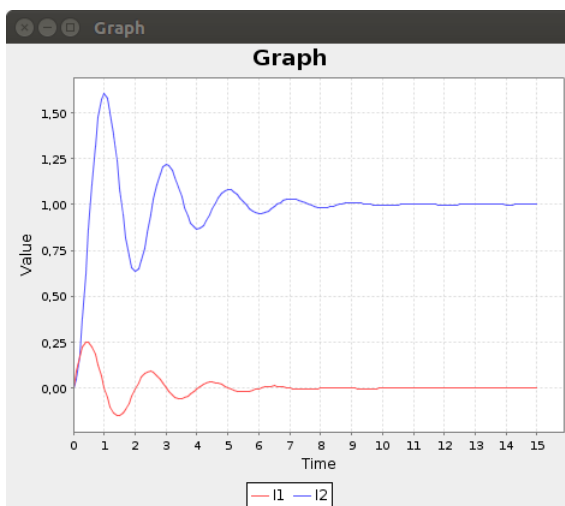
Pomocou týchto porovnávaní a pokusov sme zistili, že integračný krok a rád metódy spolu úzko súvisia, pokiaľ chceme dosahovať rovnakej presnosti výsledkov. Zvýšením integračného kroku sa nám zvýši rád metódy, a naopak. Zo zmenou integračného kroku a rádu metódy súvisí aj čas potrebný na výpočet. Keďže Moderná metóda Taylorovej rady pridáva ďalšie členy (teda vyššie derivácie a tým pádom presnejšie výsledky) v danom integračnom kroku bez nutnosti vypočítať znova celý integračný krok, je časovo výhodnejšie aby sme zvolili vyšší rád metódy ako nižší integračný krok.

## 6.7 Vplyv zmeny parametrov modelu riadenia

Vezmime si schému 6.3 a ako vstup jednosmerný prúd  $u = 1$ . Pri zmene vstupných hodnôt komponentov, napr. kondenzátor, si vieme v tejto malej schéme predstaviť, aké následky táto zmena spôsobí. Vezmime si teda napríklad kondenzátor, ktorej hodnotu zvýšime na  $C = 10F$ , čo zapríčiní pomalé nabíjanie kondenzátora 6.6. Ak hodnotu cievky znížime na  $C = 0.1F$ , v našom prípade to spôsobí rýchle rozkmitanie a postupné vyhladenie signálu 6.7. V našej schéme 6.3 sa zmena hodnoty kondenzátoru prejaví na násobičke  $N4$ , kde pri zvýšení hodnoty kondenzátora na  $C = 10F$  bude mať  $N3$  hodnotu 0.1 a pri znížení na  $C = 0.1F$  hodnotu 10 a to kvôli tomu, že násobička  $N3$  nám vyjadruje vzťah  $1/C$ . Na zmenu hodnôt v rozsiahlejších schémach nám môže pomôcť aplikácie BPAN.



Obr. 6.6: Zvysenie hodnoty kondenzátora na 10F



Obr. 6.7: Znizenie hodnoty kondenzátora na 0.1F



## 6.8 Porovnanie s Matlabom

V Matlabe sme využili funkciu ode45 a porovnali sme niekoľko sústav diferenciálnych rovníc v rýchlosti výpočtu.

$$\begin{aligned} il1' &= u - il1 - uc \\ il2' &= uc \\ uc' &= il1 - 0.5 * uc - il2 \end{aligned} \quad (6.10)$$

$$\begin{aligned} il1' &= (u - uc - 0.01 * il1) / 8 \\ il2' &= uc \\ uc' &= il1 - il2 - uc \end{aligned} \quad (6.11)$$

$$\begin{aligned} il' &= u - il1 - uc \\ uc' &= il1 - 0.5 * uc \end{aligned} \quad (6.12)$$

$$\begin{aligned} il1' &= u - il1 * il2 \\ il2' &= 0.5 * il1 * il2 \end{aligned} \quad (6.13)$$

	6.10	6.11	6.12	6.13
Matlab/ode45	5.13ms	6.89ms	6.01ms	7.13ms
BPAN/MMTR	17.35ms	83.4ms	17.3ms	28.3ms

Tabuľka 6.4: Porovnanie rýchlosti výpočtu v Matlabe a v aplikácii BPAN

MMTR - Moderná metóda Taylorovho radu.

Ako vidíme ode45 metóda v Matlabe počíta rovnice rýchlejšie. Tento výsledok môže ovplyvniť zmena veľkosti integračného kroku, ktorú ode45 využíva. Kvôli tejto zmene sme nemohli porovnať presnosť výsledkov v jednotlivých krokoch a podľa toho nastaviť parametre v aplikácii BPAN, tým pádom sme parametre zvolili čo najvhodnejšie vzhľadom na rovnakú presnosť výsledkov s ode45. Aj keď nám tieto okolnosti zapríčinili časové sklzy pri výpočtoch, vidíme, že implementovaný algoritmus v aplikácii BPAN je vhodný na využívanie.

## Kapitola 7

### Záver

V tejto práci sme sa zoznámili so systémom v reálnom čase a analyzovali sme si diferenciálne rovnice. Objasnili sme si čo je numerické derivovanie a predložili sme si numerické metódy, ktoré sa používajú na výpočet diferenciálnych rovníc, predstavili sme si Modernú metódu Taylorovho radu a jeho výpočet.

Aplikácia BPAN obsahuje dve časti. V prvej časti nám vypíše sústavu diferenciálnych rovníc z elektrického obvodu. V druhej časti nám vypíše sústavu diferenciálnych rovníc schémy modelu riadenia v reálnom čase. Diferenciálne rovnice vypočíta a výsledky zobrazí v grafe. Na výpočet rovníc využíva algoritmus Modernej metódy Taylorovho radu. Následne sme analyzovali jednotlivé vplyvy, ktoré môžu skresliť výsledky výpočtu.

Problematika výpočtu diferenciálnych rovníc numerickými metódami je tak rozsiahla, že je možné v tejto práci ďalej pokračovať a jednotlivé metódy porovnávať. Taktiež porovnávať jednotlivé vplyvy na tieto metódy, vzhľadom na presnosť a čas strávený výpočtom. V aplikácii BPAN je možné ďalej pokračovať v prepojení integrátorového obvodu a elektrického obvodu. Do elektrického obvodu by ďalšia verzia mohla priniesť zapojenie hviezdastrojuhoľník. Takto zhotovená aplikácia by sa dala využiť na výučbové účely v predmetoch zaoberajúcich sa elektrickými obvodmi (IEL, INC) a diferenciálnymi rovnicami (VNV).

# Literatúra

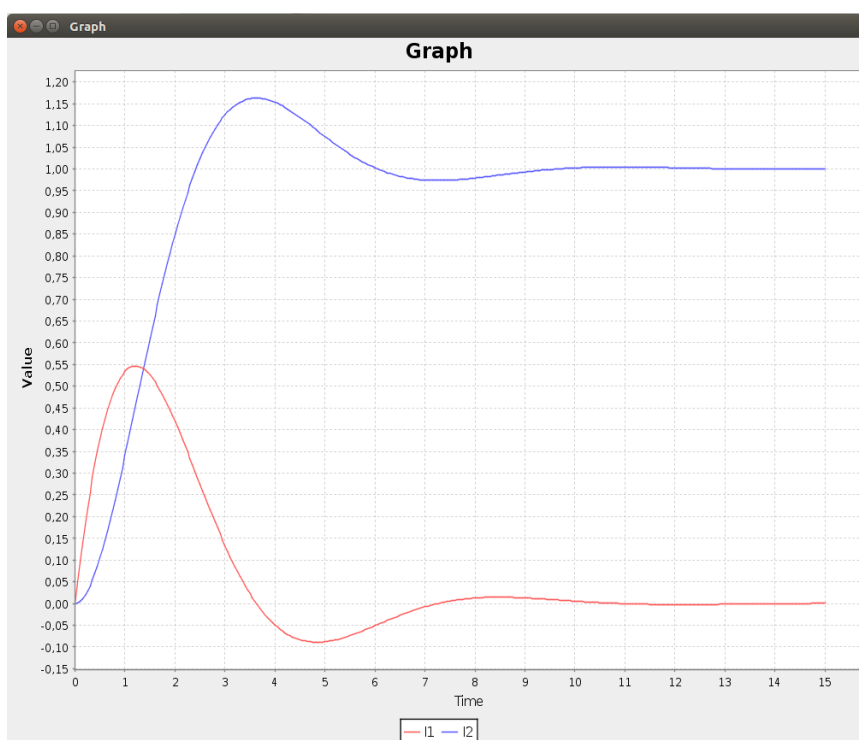
- [1] <https://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html>, [Online; navštívené 13.5.2017].
- [2] FOS WEBGUI. [http://www.fit.vutbr.cz/~kunovsky/fos/index\\_cs.php#](http://www.fit.vutbr.cz/~kunovsky/fos/index_cs.php#), [Online; navštívené 15.5.2017].
- [3] Netbeans IDE. <https://netbeans.org/>, [Online; navštívené 10.5.2017].
- [4] TKSL software: High Performance Computing. <http://www.fit.vutbr.cz/~kunovsky/TKSL/index.html.en>, [Online; navštívené 11.5.2017].
- [5] Úvod do teórie PDR. [http://mathonline.fme.vutbr.cz/download.aspx?id\\_file=787](http://mathonline.fme.vutbr.cz/download.aspx?id_file=787), [Online; navštívené 3.5.2017].
- [6] Blaha, P.; Vavřín, P.: Řízení a regulace I. [http://www.uamt.feec.vutbr.cz/~richter/vyuka/0809\\_BRR1/texty/brr1.pdf](http://www.uamt.feec.vutbr.cz/~richter/vyuka/0809_BRR1/texty/brr1.pdf), [Online; navštívené 4.5.2017].
- [7] Buša, Pirč, Schrötter: *Numerické metody, pravdepodobnosť a matematická štatistika*. 2006, ISBN 80-8073-632-4.
- [8] Daňo, Ostertagová: *Vybrané kapitoly z numerických metod, pravdepodobnosti a matematickej štatistiky*. Equilibria s.r.o., 2012, ISBN 978-80-8143-012-1.
- [9] Fajmon, B.; Hlavičková, I.: Matematika 3. <http://www.umat.feec.vutbr.cz/~hlavicka/skripta/matematika3.pdf>, [Online; navštívené 3.5.2017].
- [10] Fajmon, B.; Hlavičková, I.; aj.: Numerická matematika a pravdepodobnosť. [http://matika.umat.feec.vutbr.cz/inovace/texty/INM/CZ/INM\\_plna\\_verze\\_CZ.pdf](http://matika.umat.feec.vutbr.cz/inovace/texty/INM/CZ/INM_plna_verze_CZ.pdf), [Online; navštívené 3.5.2017].
- [11] Franců, J.: Parciální diferenciální rovnice. 2003, [Online; navštívené 3.5.2017].
- [12] Gabriela Nečasová: *Paralelní numerické řešení parciálních diferenciálních rovnic*. Diplomová práce, Brno: Vysoké učení technické v Brne – Fakulta informačních technologií, 2014.
- [13] Kunovský J.: *Modern Taylor Series Method*. Habilitačná práca, VUT v Brne, 1995.

- [14] Laplante, P. A.: *Real-time Systems design and analysis*. IEE, 2004, ISBN 0-471-22855-9.
- [15] Laplante, P. A.: *Matematika II ve strukturovaném studiu*. Praha, 2005, ISBN 80-7080-555-2.
- [16] Michal Kraus: *Paralelné výpočetné architektúry založené na numerickej integrácii*. Dizertačná práca, Brno: Vysoké učenie technické v Brne – Fakulta informačných technológií, 2013.
- [17] Pekar, L.; Kureckova, E.: Rational Approximations for Time-Delay Systems: Case Studies. [https://www.researchgate.net/profile/Libor\\_Pekar/publication/262392699\\_Rational\\_approximations\\_for\\_time-delay\\_systems\\_case\\_studies/links/54feba8b0cf2eaf210b3ba26.pdf](https://www.researchgate.net/profile/Libor_Pekar/publication/262392699_Rational_approximations_for_time-delay_systems_case_studies/links/54feba8b0cf2eaf210b3ba26.pdf), [Online; navštívené 13.5.2017].
- [18] Peringer, P.: Modelování a simulace. <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIMS-IT%2Ftexts%2Fopora-ims.pdf&cid=11453>, 2012, [Online; navštívené 3.5.2017].
- [19] Petr Veigend: *MODELOVÁNÍ A ŘÍZENÍ V REÁLNÉM ČASE*. Pojednání k tématu disertační práce, Brno: Vysoké učenie technické v Brne – Fakulta informačných technológií, 2016.
- [20] Ralston, A.: *Základy numerické matematiky*. Academia, 1978.
- [21] Richard L. Burden; J.Douglas Faires: *Numerical Analysis*. Brooks/Cole, 2011, ISBN 978-0-538-73351-9.
- [22] Václav Ibl: *Taylorovy řady elementárních funkcí*. Diplomová práce, Praha: Univerzita Karlova v Praze – Matematicko-fyzikální fakulta, 2009.

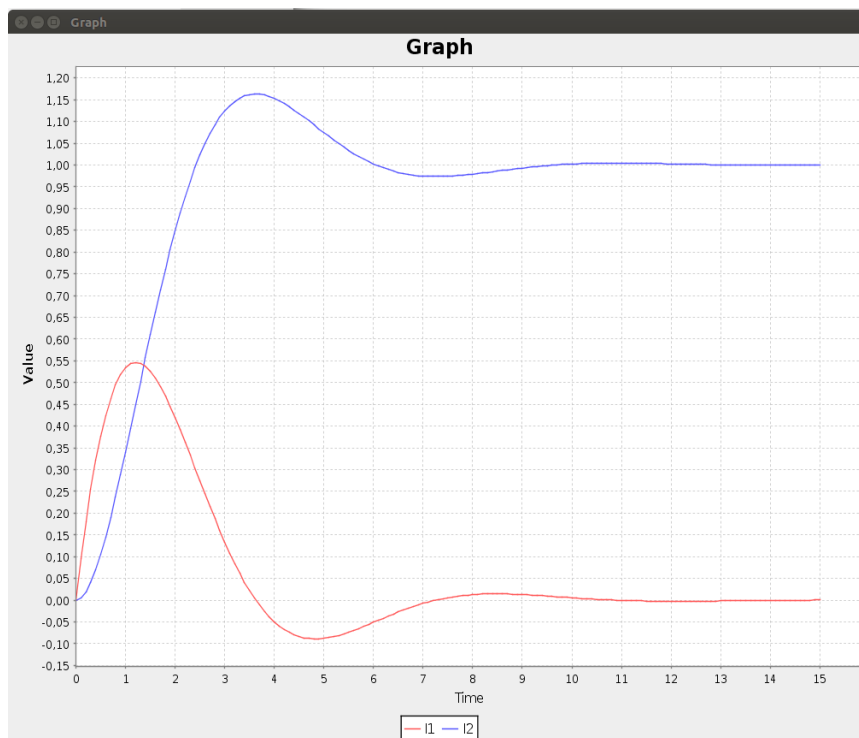
# Prílohy

## Príloha A

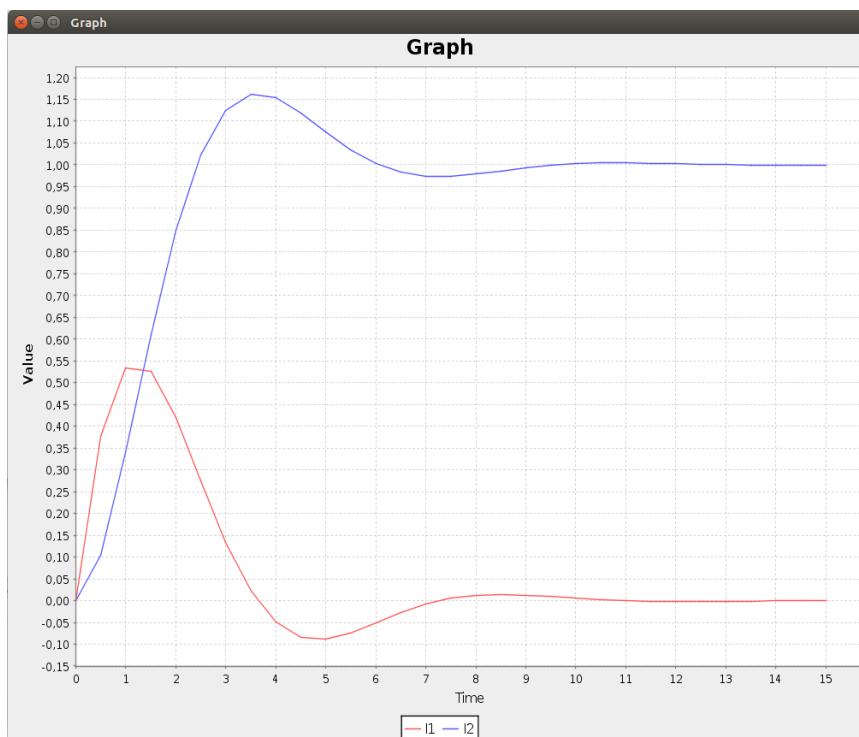
# Kvantovanie



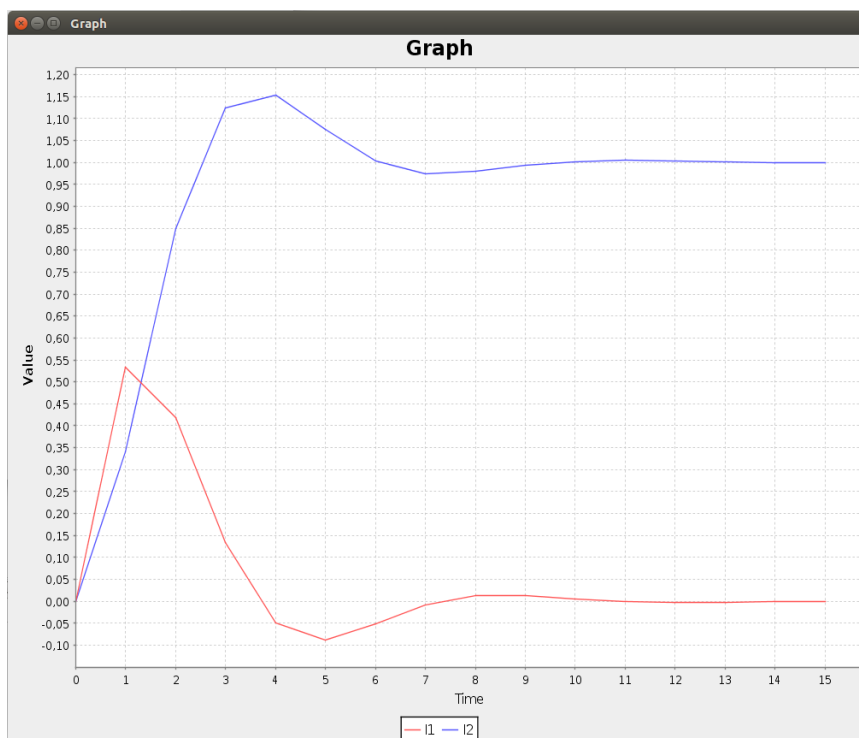
Obr. A.1: Vplyv kvantovania krok = 0.05



Obr. A.2: Vplyv kvantovania krok = 0.1



Obr. A.3: Vplyv kvantovania krok = 0.5

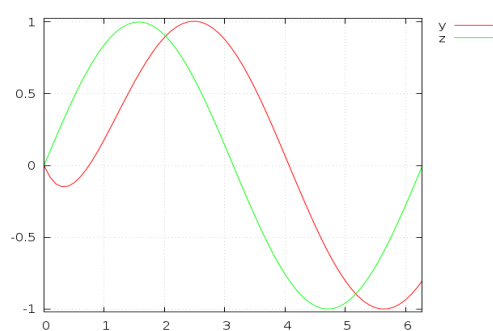


Obr. A.4: Vplyv kvantovania krok = 1

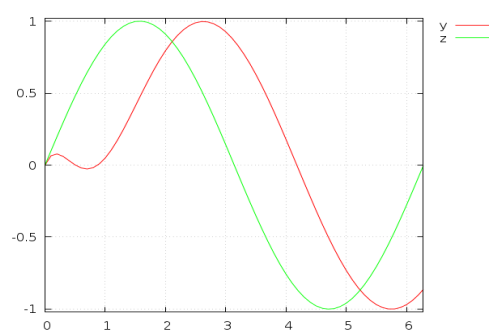


## Príloha B

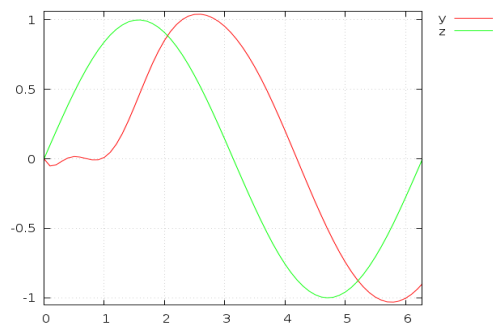
# Padeho rozvoj



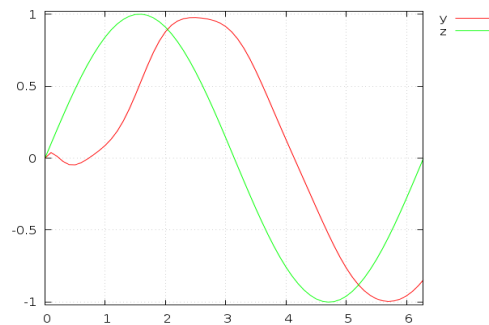
(a)  $n = 1$



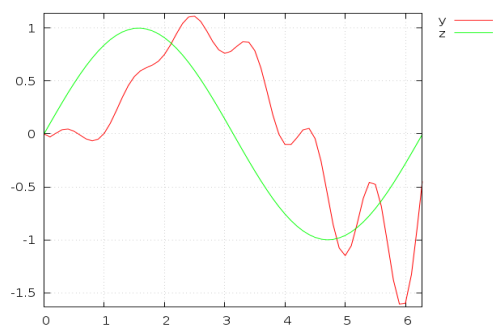
(b)  $n = 2$



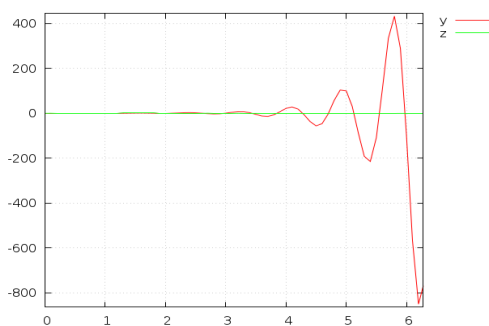
(c)  $n = 3$



(d)  $n = 4$

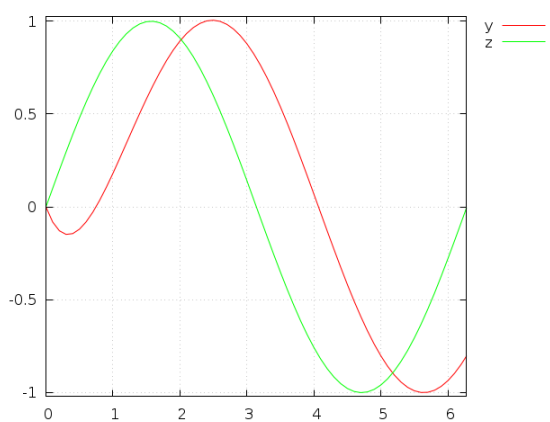


(e)  $n = 5$

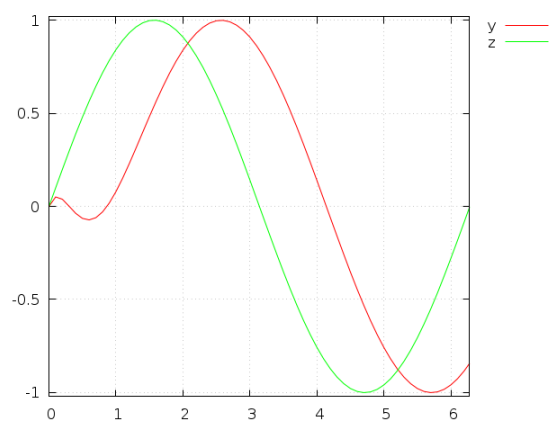


(f)  $n = 6$

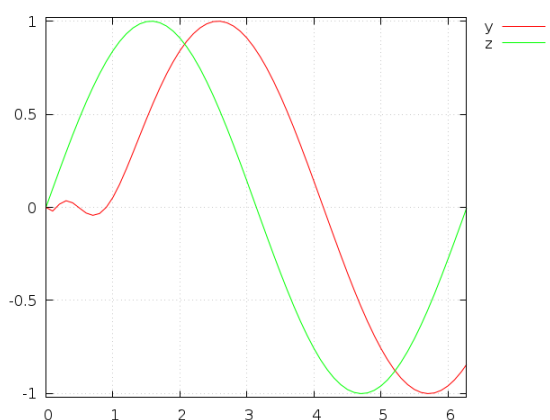
Obr. B.1: Padeho aproximácia rádu  $n$  v programe FOS



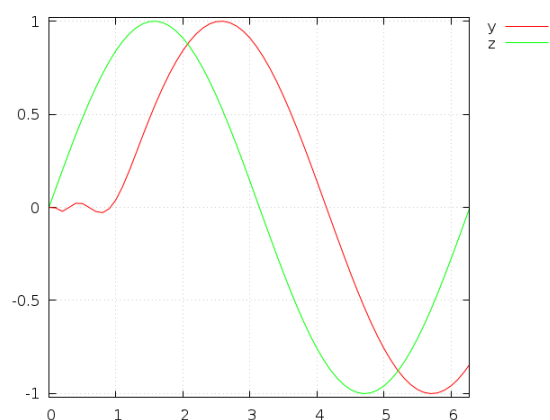
(a)  $n = 1$



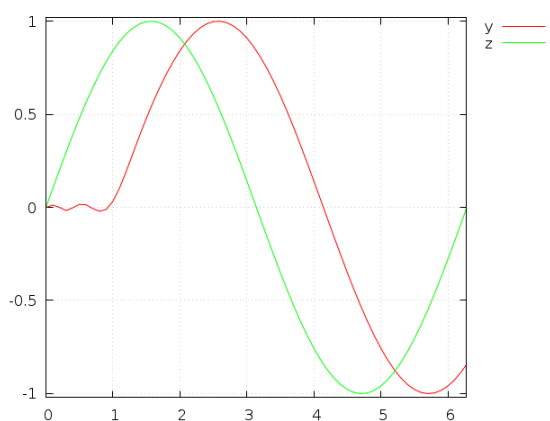
(b)  $n = 2$



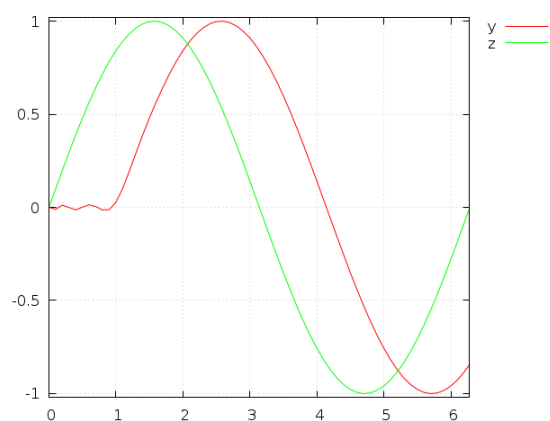
(c)  $n = 3$



(d)  $n = 4$



(e)  $n = 5$



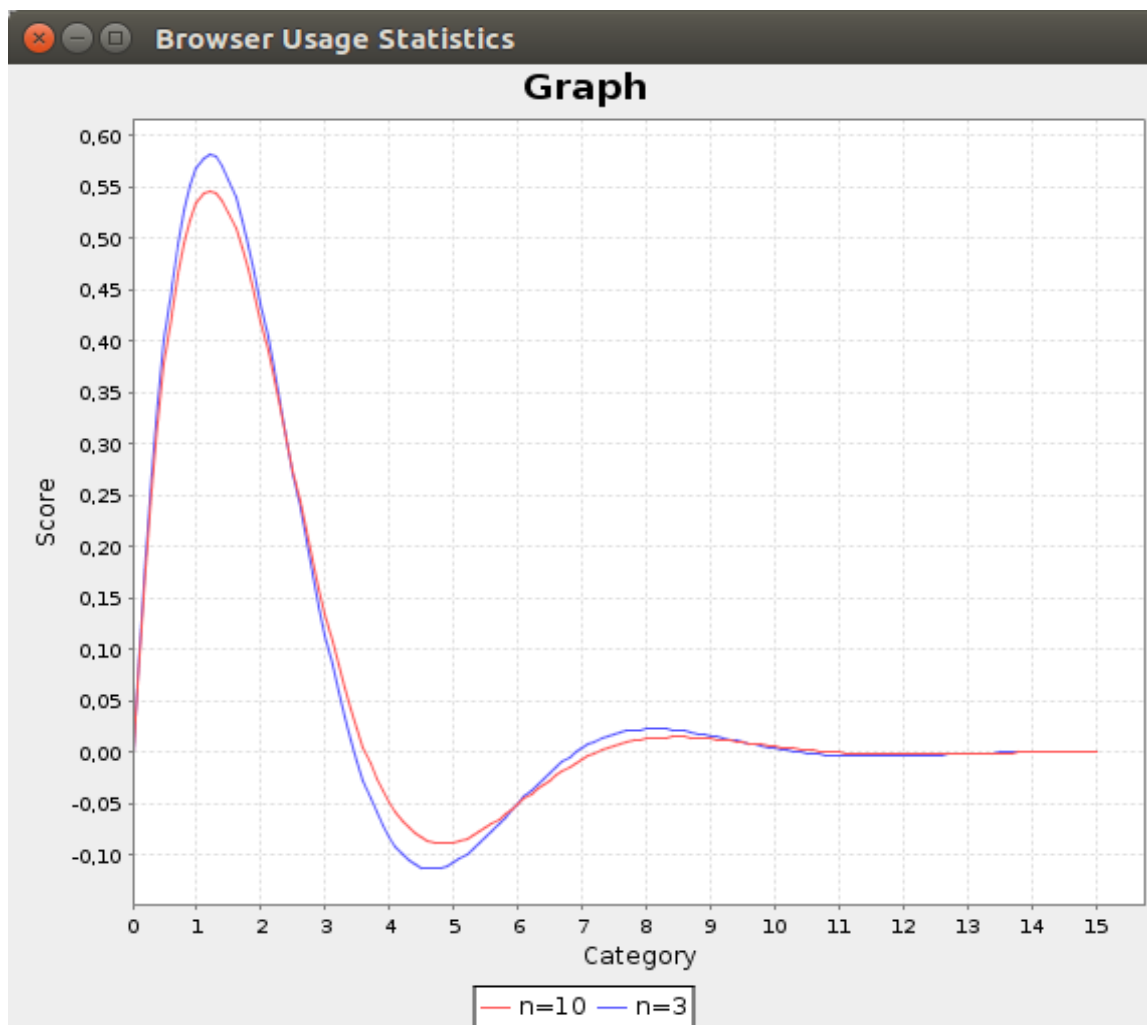
(f)  $n = 6$

Obr. B.2: Diagonálna Pahedo aproximáciá rádu  $n$  v programe FOS

## Príloha C

# Rád metódy

Kvôli prehľadnosti grafu, a tým pádom zreteľnosti rozdielu výsledkov, sme v grafe použili len dve hodnoty. Rozdielnosť výsledkov sme skúšali pri parametri  $\text{maxORD} = 10, 8, 5$  a  $3$ . Aby boli rozdieli čo najviac viditeľné sme zvolili hodnoty  $\text{maxORD}$ , ktoré sú od seba najviac odlišné, čiže  $10$  a  $3$ .



Obr. C.1: Rozdiel výsledkov pri rádu  $n$

## Príloha D

# Obsah CD

Priložené CD obsahuje:

- Text práce vo formáte pdf.
- Zdrojové súbory práce v  $\text{\LaTeX}$ u
- Zdrojové súbory aplikácie BPAN v jazyku Java
- Spustiteľný súbor aplikácie BPAN - BPAN.jar